

Desarrollo de una aplicación en Android para la estimación automática de carbohidratos y nutrientes alimenticios mediante captura y análisis de imágenes aplicando algoritmos de Inteligencia artificial.

Cristeam Caiola Pasquier

**GRADO EN INGENIERÍA DEL SOFTWARE
FACULTAD DE INFORMÁTICA
DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID**



TRABAJO DE FIN DE GRADO EN INGENIERÍA DEL SOFTWARE

Madrid, 13 de septiembre del 2016

Director | José Ignacio Hidalgo Pérez

Colaboradores | Maria Guijarro Mata-García y Gonzalo Pajares Martinsanz



AUTORIZACIÓN PARA LA DIFUSIÓN DEL TRABAJO FIN DE GRADO Y SU DEPÓSITO EN EL REPOSITORIO INSTITUCIONAL E-PRINTS COMPLUTENSE

Los abajo firmantes, alumno/s y tutor/es del Trabajo Fin de Grado (TFG) en el Grado en de la Facultad de, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TF) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Periodo de embargo (opcional):

☐ 24 meses

TÍTULO del TFG:
.....

Curso académico: 20..... / 20.....

Nombre del Alumno/s:

.....

Tutor/es del TFG y departamento al que pertenece:

.....

Firma del alumno/s

Firma del tutor/es



Este documento se distribuye bajo **licencia Creative Commons BY-NC-SA 3.0**.

Dicha licencia permite:

Compartir — copiar y redistribuir el material en cualquier medio o formato.

Adaptar — transformar y crear a partir del material

Bajo las condiciones siguientes:

Reconocimiento — Debe reconocer la autoría del documento. Debe proporcionar un enlace a este documento original e indicar si se han realizado modificaciones sobre el mismo.

No Comercial — No puede utilizar el material para una finalidad comercial.

Compartir Igual — Si transforma o crea a partir del material, deberá difundir sus contribuciones bajo la misma licencia que el original.

A mi familia

Por su apoyo, fundamental para lograr mis metas.

Agradecimientos

Mis más sinceros agradecimientos al director de proyecto **José Ignacio Hidalgo** por todo el tiempo dedicado, por el interés mostrado y por su paciencia y amabilidad durante el desarrollo del proyecto.

Agradecimientos A la profesora y tutora **Maria Guijarro** por sus aportaciones claves y concisas para el desarrollo de la aplicación.

Agradecimientos al profesor y colaborador **Gonzalo Pajares** por su siempre inmediata disposición para ayudar y por mostrarme el fascinante mundo de la Inteligencia Artificial.

A mis compañeros **Abdellah Absaoui** y **Francisco Linde** por su amistad a lo largo de la carrera en la facultad y colaboración para la versión en iPhone del proyecto.

Índice

Contenido

Índice de figuras.....	XIII
Índice de abreviaturas	XIV
Resumen	XVI
Palabras clave	XVI
Abstract.....	XVII
Keywords.....	XVII
Capítulo 1. Introducción.....	1
1.1. Antecedentes	1
• Diabetes	1
• glUCModel & Android App.....	2
1.3. El proyecto: Objetivos y plan de trabajo.....	3
1.4. Motivación Personal	3
Chapter 1. Introduction	5
1.1. Background	5
• Diabetes	5
• glUCModel & Android App.....	6
1.3. Project Objectives and work plan	7
1.4. Personal Motivation.....	7
Capítulo 2. Desarrollando en Android	9
2.1. ¿Por qué Android?	9
Capítulo 3. Implementación.....	11
Fase 1. Generación de la base del conocimiento para los algoritmos de IA	11
Fase 1.A.	11
Fase 1.B.	13
Fase 2. Entrenamiento de los algoritmos de IA	16
Fase 3. Generar la base de datos con la información nutricional para cada alimento y su salvado en el servidor API	19
Fase 4. Creación del servidor de aplicaciones (API) y el cliente (Android App).....	21

Fase 4.A. Creación del Servidor API	21
Fase 4.B.1. Creación del Cliente (APP Android)	23
Fase 4.B.2. Parametrización del software base para Android	25
Fase 4.B.3. Construcción del sistema.....	26
Fase 4.B.4. Concurrencia, paralelismo e Hilos	28
WEB DEMO	30
Capítulo 4. Análisis de Ingeniería del Software aplicados al proyecto	31
4.1. Análisis de rendimiento de la App	31
4.2. Análisis de riesgos	34
4.3. Análisis de Costes, fortalezas, oportunidades, debilidades y amenazas	37
4.3.1. Análisis de costes	37
4.3.2. Fortalezas	37
4.3.3. Oportunidades	37
4.3.4. Debilidades.....	38
4.3.5. Amenazas	38
4.4. Análisis de Mantenibilidad, Seguridad, Disponibilidad, Usabilidad	38
4.4.1. Mantenibilidad.....	38
4.4.2. Seguridad	38
4.4.3. Disponibilidad	38
4.4.4. Usabilidad.....	39
Capítulo 5. Manual de usuario.....	41
Menú principal	41
Usuario realiza foto (Ejemplo para un Aguacate).....	42
Confirmación que la foto ha sido tomada y que los pixeles han sido analizados.....	43
Se muestran posibles alimentos coincidentes.....	44
Se muestra mensaje de espera mientras se cargan los detalles	45
Se muestran las propiedades nutricionales del alimento seleccionado, teniendo el usuario la opción de editar el campo de la porción ingerida en gramos, actualizándose todos los demás valores en base a esta.....	46
Usuario guarda los datos del alimento y la cantidad ingerida en su registro personal.....	47
En caso que no existan coincidencias, el usuario tiene la opción de agregar un nuevo alimento, para ello se hace uso de la API de USDA la cual devuelve los siguientes alimentos para el ejemplo de aguacate:.....	48
Usuario selecciona y guarda alguno de los resultados en la BD	49

Capítulo 6. Conclusiones y futuro de la aplicación	51
Chapter 6. Conclusion and future	53
Bibliografia	55

Índice de figuras

<i>Ilustración 1: Tasa de mercado de Android y demás sistemas operativos</i>	<i>9</i>
<i>Ilustración 2: Distribución de las diferentes versiones de Android</i>	<i>10</i>
<i>Ilustración 3: Esquemas de las fases del proyecto</i>	<i>11</i>
<i>Ilustración 4: Almacenamiento por meses en Google Drive de las imágenes que compondrán la base del conocimiento.....</i>	<i>12</i>
<i>Ilustración 5: Diferentes menús diarios fotografiados.....</i>	<i>12</i>
<i>Ilustración 6: Ejemplo de Menús diarios para el 21 de mayo del 2015.....</i>	<i>13</i>
<i>Ilustración 8: IDE MatlabR2015b</i>	<i>14</i>
<i>Ilustración 7: 50 muestras iniciales.....</i>	<i>14</i>
<i>Ilustración 9: Filtros aplicados a la imagen para extraer los pixeles del área seleccionada.....</i>	<i>15</i>
<i>Ilustración 10: Archivos generados de la región seleccionada de la imagen muestra</i>	<i>15</i>
<i>Ilustración 11: Formula de la distancia de Mahalanobis usada para calcular el grado de semejanza entre diferentes muestras</i>	<i>17</i>
<i>Ilustración 12: Clases Bayes de IA</i>	<i>17</i>
<i>Ilustración 13: IDE Eclipse generando la data IA.....</i>	<i>18</i>
<i>Ilustración 14: BD generada con IA data de cada alimento.....</i>	<i>18</i>
<i>Ilustración 15 Valores nutricional de los alimentos en la BD.....</i>	<i>19</i>
<i>Ilustración 16: Información nutricional de los alimentos en la BD</i>	<i>20</i>
<i>Ilustración 17: Esquema y detalle de la BD</i>	<i>20</i>
<i>Ilustración 18: Esquema general del patrón MVC.....</i>	<i>21</i>
<i>Ilustración 19: Patrones IS usados en el servidor API.....</i>	<i>22</i>
<i>Ilustración 20: Arquitectura del proyecto</i>	<i>23</i>
<i>Ilustración 21: Definición gráfica de los hilos multi tarea en el SO.....</i>	<i>28</i>
<i>Ilustración 22: Ejemplo de la implementación de hilos en Android</i>	<i>29</i>
<i>Ilustración 23: Demo WEB</i>	<i>30</i>
<i>Ilustración 24: Rendimiento con programa monitor A antes de ejecutar App</i>	<i>31</i>
<i>Ilustración 25: Rendimiento con programa monitor B antes de ejecutar App</i>	<i>32</i>
<i>Ilustración 26: Rendimiento con programa monitor A despues de ejecución.....</i>	<i>32</i>
<i>Ilustración 27: Rendimiento con programa monitor B después de ejecución.....</i>	<i>33</i>
<i>Ilustración 28: App menú principal</i>	<i>41</i>
<i>Ilustración 29: Captura de foto</i>	<i>42</i>
<i>Ilustración 30: Mensajes de confirmación en la App</i>	<i>43</i>
<i>Ilustración 31: App muestra alimentos coincidentes</i>	<i>44</i>
<i>Ilustración 32: Mensajes de espera en la App.....</i>	<i>45</i>
<i>Ilustración 33: Propiedades nutricionales del alimento</i>	<i>46</i>
<i>Ilustración 34: Grabado de datos en la BD del usuario.....</i>	<i>47</i>
<i>Ilustración 35: Conexión de la App con Api externa.....</i>	<i>48</i>
<i>Ilustración 36: Salvado de nuevo alimento en la BD.....</i>	<i>49</i>
<i>Ilustración 37: Algoritmo Sobel.....</i>	<i>52</i>

Índice de abreviaturas

- **TFG:** Trabajo de Fin de Grado.
- **IA:** Inteligencia Artificial.
- **APP:** Aplicación para móviles (Smartphone).
- **BBDD:** Bases de datos.
- **IDE:** (Entorno de desarrollo integrado) entorno de ayuda al programador para desarrollar las aplicaciones.
- **API:** (Interfaz de programación de aplicaciones) método de comunicación entre el móvil y el servidor donde se procesan los datos.
- **CH:** Carbohidratos.
- **SO:** Sistema operativo.
- **RGB:** Red, Green, Blue (cantidad de rojo, verde y azul que componen cualquier imagen digital).
- **glUCModel:** Macro proyecto y plataforma de ayuda a pacientes con diabetes.
- **CESNID:** Centre d'Ensenyament de Nutrició Humana i Dietètica de la Universitat de Barcelona.
- **USDA:** Departamento de agricultura de los estados unidos.
- **IS:** Ingeniería del Software.
- **MVC:** (Modelo Vista Controlador) Patrón de IS para gestionar proyectos.
- **CPU:** Central Processing Unit, procesador/es de cualquier dispositivo que realice cálculos computacionales.
- **RAM:** Random Access Memory, tipo de memoria que almacena datos temporales para poder ejecutar la programas.

- **HTML:** HyperText Markup Language, lenguaje de etiquetas, para programar webs.
- **CSS:** Cascading Style Sheets, lenguaje utilizado para aportar estilo a las páginas HTML.
- **CSV:** Comma-Separated Values, tipo de formato de archivo para hojas de cálculo.
- **FTP:** File Transfer Protocol, protocolo de transferencia de archivos
- **SDK:** Software Development Kit

Resumen

Este trabajo tiene como meta el desarrollo de una App para Android capaz de realizar una foto y reconocer el alimento que está en la misma para luego mostrar:

- La información nutricional del alimento (haciendo énfasis en los carbohidratos).
- Mostrarle al paciente cual es la ración recomendada de dicho alimento para su dieta.
- Grabar toda esta información (siempre proporcional a la cantidad ingerida), en el histórico del paciente.

Para este último punto se realizara la integración de esta App con la plataforma **glUCModel** la cual lleva el histórico de los pacientes junto con sus datos.

El objetivo Final es el de ayudar al paciente diabético en su control diario de la enfermedad, proporcionando una herramienta de estimación de carbohidratos y nutrientes fiable.

Palabras clave

Píxeles, RGB, Diabetes, fórmulas de IA, algoritmo, Bayes, Lenguajes de programación, muestras.

Abstract

This work is aimed at developing an app for Android able to take a picture and recognize the food inside it and show:

- The information nutrition food (with emphasis on carbohydrates).
- Show to the patient which is the recommended food ration for their diet.
- Record all this information (always proportional to the amount of food ingested), in the patient history.

For this last point the integration of this app is held with the platform glUCModel which carries the history of the patients along with other data.

The Final aim is to help the diabetic patient in their daily control of the disease, providing a reliable tool with which estimate carbohydrates and nutrients.

Keywords

Pixels, RGB, Diabetes, AI formulas, algorithm, Bayes, Programming languages, samples.

Capítulo 1. Introducción

1.1. Antecedentes

• Diabetes

La Diabetes Mellitus es una enfermedad que afecta a más de 350 millones de personas en todo el mundo. Se caracteriza porque hay un defecto en la acción o en la generación de insulina por parte del páncreas, lo que hace que la glucosa no sea asimilada por las células y se incremente la concentración de glucosa en sangre (hiperglucemia).[1]

La mayoría de los diabéticos necesitan hacer un aporte exógeno de insulina para controlar los niveles de glucosa en sangre. Para establecer la cantidad de insulina necesaria, una de las cosas que tienen que hacer es una estimación de los carbohidratos que contiene el alimento que va ingerir.

Según un estudio realizado por di@bet.es en 2011 el 13,8% de los españoles mayores de 18 años sufre diabetes, y a nivel mundial hay más de 260 millones de personas diabéticas. Estos datos arrojan suficiente luz acerca de la magnitud del problema [15].

La asociación americana de diabetes, The American Diabetes Association, clasifica la diabetes en 2 tipos:

- Tipo 1: El páncreas del paciente no genera insulina, de tal modo que los niveles de glucosa no pueden ser reducidos adecuadamente.

Estos pacientes son insulino dependientes, es decir requieren de un aporte externo de insulina en su día a día. La forma de incorporar este aporte insulínico puede ser mediante una bomba o mediante inyecciones.

- Tipo 2: El paciente puede generar insulina en dosis inferiores a las correctas y en ocasiones a esto se le debe añadir que el cuerpo muestra resistencia a la insulina, lo cual la convierte en menos efectiva aumentando los niveles de glucosa en sangre. Estos pacientes no suelen ser insulino dependientes en las etapas iniciales de la enfermedad aunque suelen llegar a serlo .

Afortunadamente más del 90% de los diabéticos se encuentran en el segundo grupo, y decimos afortunadamente porque este tipo de diabetes es menos incapacitante que la diabetes tipo 1. El principal tratamiento para la diabetes tipo 2 consiste en lograr un peso saludable y mantener una alimentación sana, pero en algunos casos se requiere tratamientos en forma de pastillas o en etapas avanzadas se necesitan también un aporte externo de insulina.

Los diabéticos tipo 1 son por tanto los diabéticos que ven su vida afectada en mayor medida. Deben mantener un control meticuloso de sus niveles de glucemia en sangre para prevenir tanto hiperglucemias (niveles $> 120\text{mg/dl}$) como hipoglucemias (niveles $< 40\text{mg/dl}$). El control consiste en hacer mediciones de glucemia en sangre, para realizar posteriores predicciones basadas en la ingesta de comida que van a realizar, el posible deporte que vayan a practicar, el peso que tengan en ese momento y muchos otros factores que influyen en los niveles de glucemia. En base a esta predicción deben realizar una estimación de la insulina que van a inyectarse para mantener sus niveles de glucemia en unos valores adecuados.

• glUCModel & Android App

La tarea de un diabético tipo 1 no es fácil. En la mayoría de los casos los pacientes perfeccionan sus predicciones con la práctica, y por supuesto, con la ayuda de los profesionales de la salud, pero aún con muchos años de experiencia el paciente yerra en sus estimaciones. Es en esta tarea diaria en la que glUCModel y esta App quieren jugar un papel relevante. La idea detrás de todo el macro proyecto es la de ser capaces de predecir los niveles en sangre del paciente basándose en parámetros actuales del sujeto, como son el nivel de azúcar o la próxima ingesta de carbohidratos con el fin de aconsejarle una cantidad de insulina determinada.

Para realizar los cálculos glUCModel está utilizando técnicas de inteligencia artificial como las gramáticas evolutivas, basadas en programación genética. Este tipo de orientación permite una mayor libertad a la hora de definir la expresión matemática de la solución, puesto que no se tienen restricciones como por ejemplo se pueden encontrar utilizando ecuaciones lineales. El proyecto lleva varios años desarrollándose mediante la colaboración de la Universidad Complutense de Madrid, el Hospital Universitario Príncipe de Asturias (Alcalá de Henares, España) y el Hospital Virgen de la Salud (Toledo, España). Los profesionales implicados en el proyecto son [4]:

- Ignacio Hidalgo, Juan Lanchares, Oscar Garnica, Manuel Velasco y Sergio Contador del departamento de Arquitectura y Automática de la Universidad Complutense de Madrid.
- J. Manuel Colmenar de la Universidad Rey Juan Carlos de Madrid.
- Esther Maqueda del departamento de nutrición y endocrinología del Hospital Virgen de la Salud (Toledo, España)
- Marta Botella y José Antonio Rubio del departamento de nutrición y endocrinología del hospital Universitario Príncipe de Asturias (Alcalá de Henares, España).

En definitiva glUCModel junto con esta App permitirán:

- Mejorar el control de la diabetes.

- Ayudar a los diabéticos y a sus médicos a controlar la enfermedad.

1.3. El proyecto: Objetivos y plan de trabajo.

El objetivo de este trabajo es el desarrollo de una aplicación móvil que estime de forma automática la cantidad de carbohidratos, CH, y ayudar al paciente a tener un mejor control de su dieta.

Esta aplicación será integrada dentro del macro proyecto de glUCModel como un módulo añadido.

En cuanto a los problemas a resolver, el principal es el de moldar todo el proyecto de tal manera que los algoritmos de IA sean capaces de predecir con cierto (alto) grado de acierto los posibles alimentos y sus valores nutricionales, para ello se generó un plan de trabajo dividido en **4 fases** (detalladas en el capítulo III) cada una delimitada con específicos objetivos y problemas a resolver.

1.4. Motivación Personal

Varios fueron los motivos que me impulsaron a desarrollar este proyecto, entre los principales:

- Siempre había querido programar una aplicación para móviles ya que implican un afianzamiento de todos los conocimientos adquiridos durante la carrera y estar actualizado con las ultimas tendencias tecnológicas.
- El tener que usar técnicas dentro del paradigma de la inteligencia artificial, concretamente técnicas de visión artificial con el fin de identificar los alimentos.
- Por último, el hecho de crear un software que aporte una ayuda en el mundo sanitario.

Chapter 1. Introduction

1.1. Background

• Diabetes

Diabetes Mellitus is a disease that affects more than 350 million people worldwide. It is characterized by a defect in action or the generation of insulin by the pancreas, it makes glucose not being absorbed by the cells and the blood glucose concentration (hyperglycemia) increases.

Most diabetics need to make an exogenous supply of insulin to control blood glucose levels. To set the amount of insulin needed, one of the things they have to do is an estimate of carbohydrate containing food to be eaten.

According to a study by di@bet.es in 2011 13.8% of Spaniards over 18 years old he suffers from diabetes, and worldwide there are more than 260 million people with diabetes. These data shed enough light on the magnitude of the problem.

The American Diabetes Association classified the diabetes into 2 types [3]:

- Type 1: The pancreas patient not creates insulin, so that glucose levels cannot be reduced properly.

These patients are insulin dependent, therefore they require external supply of insulin in their daily lives. How to incorporate this insulin delivery may be by a pump or by injection.

- Type 2: The patient can generate insulin doses below the right levels and sometimes the body shows insulin resistance, which makes it less effective by increasing blood glucose levels. These patients tend not to be insulin-dependent in the early stages of the disease although they tend to become so.

Fortunately over 90% of diabetics are in the second group, and say fortunately because this type of diabetes is less disabling than type 1. The main treatment for type 2 is to achieve a healthy weight and maintain power healthy, but in some cases require treatment in pill form or in advanced stages also need external insulin supply.

Type 1 diabetics are therefore diabetics who see their lives affected to a greater extent. They must maintain a careful control of their blood glucose levels to prevent both hyperglycemia (levels > 120mg / dl) and hypoglycemia (levels <40 mg / dl). The control consists of measurements of blood glucose, to make further predictions based on food intake. Possible sport, the weight you have at the time and many other factors influencing in blood sugar levels. Based on this prediction should make an estimate of insulin to be injected to keep their blood glucose levels appropriate values.[2]

• glUCModel & Android App

The task of a Type 1 diabetic is not easy. In most cases patients refine their predictions with practice, and of course, with the help of health professionals, but still with many years of experience the patient make mistake in its estimates. It is in this daily task that glUCModel wants to play a role. The idea behind glUCModel is to be able to predict the patient's blood levels based on current parameters of the subject, such as sugar level or the next intake of carbohydrates in order to advise a certain amount of insulin.

To perform calculations glUCModel are using artificial intelligence techniques such as evolutionary grammars based on genetic programming. This type of orientation allows greater freedom in defining the mathematical expression of the solution, since you do not have restrictions such as can be found using linear equations. The project takes several years to develop through the collaboration of the Complutense University of Madrid, the Prince of Asturias University Hospital (Alcalá de Henares, Spain) and the Hospital Virgen de la Salud (Toledo, Spain). The professionals involved in the project are:

- Ignacio Hidalgo, Juan Lanchares, Oscar Garnica, Manuel Velasco and Sergio counter Architecture and Automation Department of the Complutense University of Madrid.
- Colmenar J. Manuel Rey Juan Carlos University in Madrid.
- Esther Maqueda the department of nutrition and endocrinology at the Hospital Virgen de la Salud (Toledo, Spain)
- Marta Bottle and José Antonio Rubio the department of nutrition and endocrinology at the University Hospital Prince of Asturias (Alcalá de Henares, Spain).

In conclusion glUCModel together this App will allow:

- Improve control of diabetes.
- Helping diabetics and their doctors to control the disease.

1.3. Project Objectives and work plan

The aim of this work is the development of a mobile application that automatically estimates the amount of carbohydrates, CH, and helps the patient to have better control of your diet.

This application will be integrated into the macro project glUCModel as an added module.

As for the problems to be solved, the main one is modulate the project so that the AI algorithms are able to predict with a certain (high) level of accuracy possible food, to achieve that a plan was generated and divided in 4 phases (detailed in chapter III) each one defined with specific objectives and problems to solve.

1.4. Personal Motivation

There were several reasons that prompted me to develop this Project:

- Always wanted to program an App for smartphones since they involve a strengthening of all the knowledge acquired during the university studies and being upgrade with the new technologies.
- Use AI computer vision techniques in order to identify the foods..
- Finally, the fact of create a software that provides an aid in the healthcare sector.

Capítulo 2. Desarrollando en Android

2.1. ¿Por qué Android?

Si se estudia el crecimiento y el uso de los sistemas operativos en Smartphone en el mercado de los últimos años, se concluye que tanto Android como IOS son los más usados.

Para ser más precisos, a finales del 2015 estos dos sistemas compartían el 96% de la cuota del mercado.

Android por su parte es el SO preferido a nivel mundial, siendo usado por el 82% del total de usuarios con Smartphone. Por este motivo he seleccionado este sistema operativo para desarrollar la aplicación.

A continuación se pueden observar estos datos gráficamente:

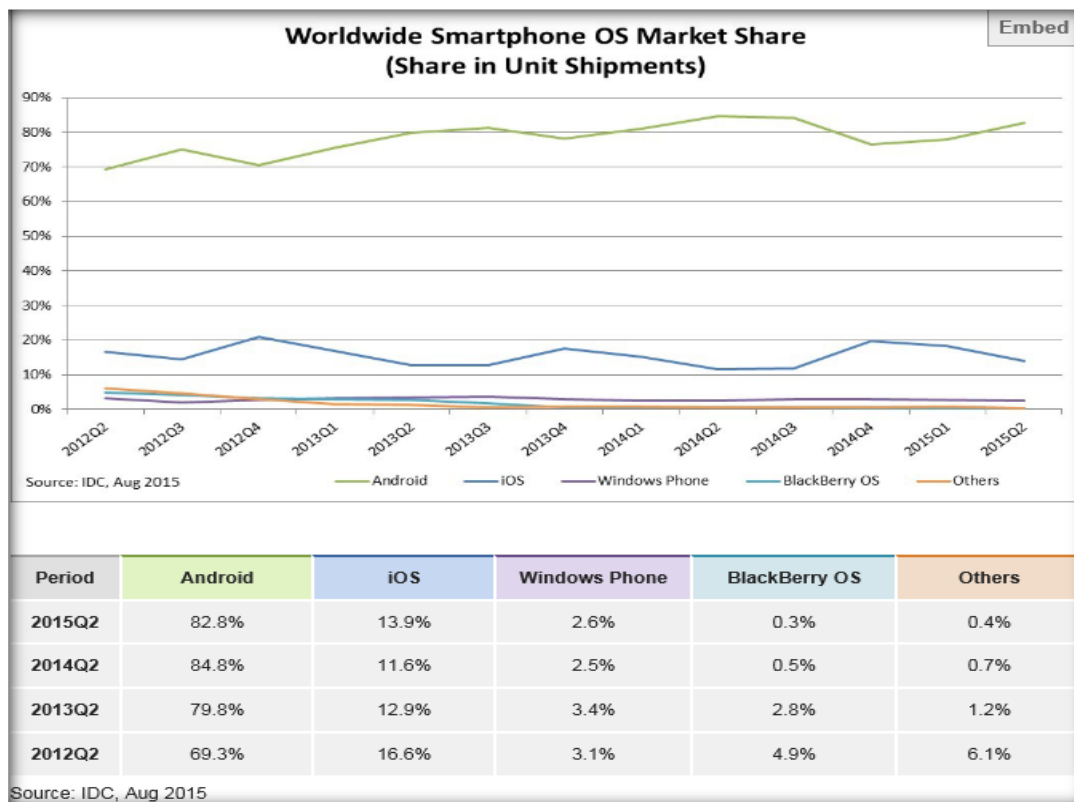


Ilustración 1: Tasa de mercado de Android y demás sistemas operativos

Fuente: IDC.com

Android, a su vez, posee varias versiones, siendo la más usada como se puede apreciar en la gráfica la versión Kika 4.4 (con la cual se programó y probó este proyecto)

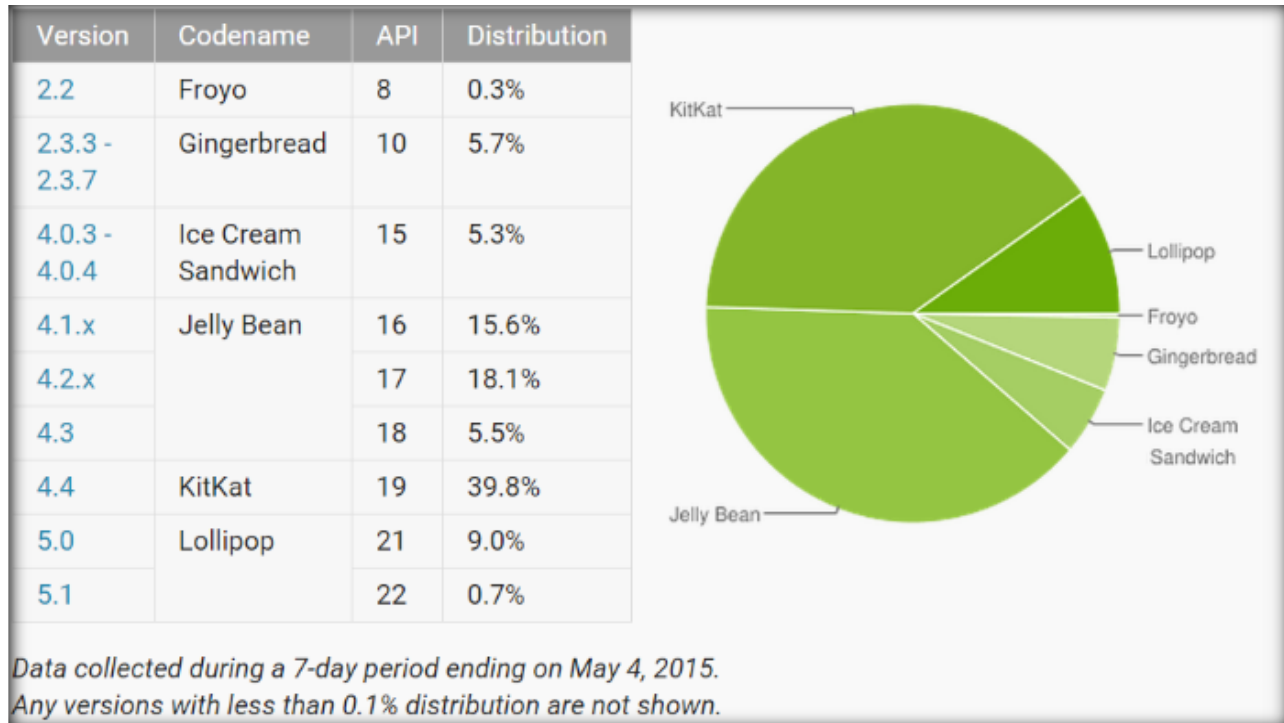


Ilustración 2: Distribución de las diferentes versiones de Android

Fuente: Daillytech.com

Capítulo 3. Implementación

Para realizar esta app se ha hecho uso de las técnicas de inteligencia artificial (IA.) estudiadas durante la carrera con el profesor G. Pajares. El proyecto fue dividido en cuatro fases:

Fase 1: Generar la base del conocimiento para los algoritmos de IA.

Fase 2: Entrenar los algoritmos para que sean capaces de saber y reconocer los alimentos.

Fase 3: Obtener la información nutricional de cada alimento y grabarla en el servidor API

Fase 4: Desarrollo de la estructura cliente (Android app) y servidor (API) para establecer la comunicación de la app.

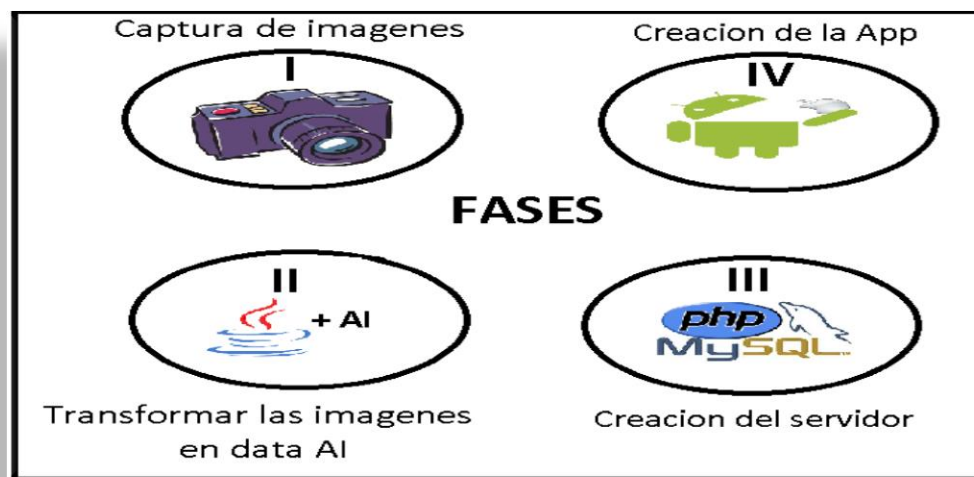


Ilustración 3: Esquemas de las fases del proyecto

Fase 1. Generación de la base del conocimiento para los algoritmos de IA

Fase 1.A.

Tecnologías usadas: Google drive, Dropbox V10.3.20

La base del conocimiento son los datos que necesitamos para empezar a modelar el comportamiento y entrenar a los algoritmos, en nuestro caso a estos datos se les aplicaran fórmulas de Bayes de IA las cuales nos permitirán identificar los alimentos por medio de matrices y vectores.

Esta base de datos la forman 574MB en fotografías de muestras de alimentos tomadas durante los meses de marzo a junio del año 2015 en la cafetería de la Facultad de Informática de la UCM sobre los menús diarios que se ofrecían..

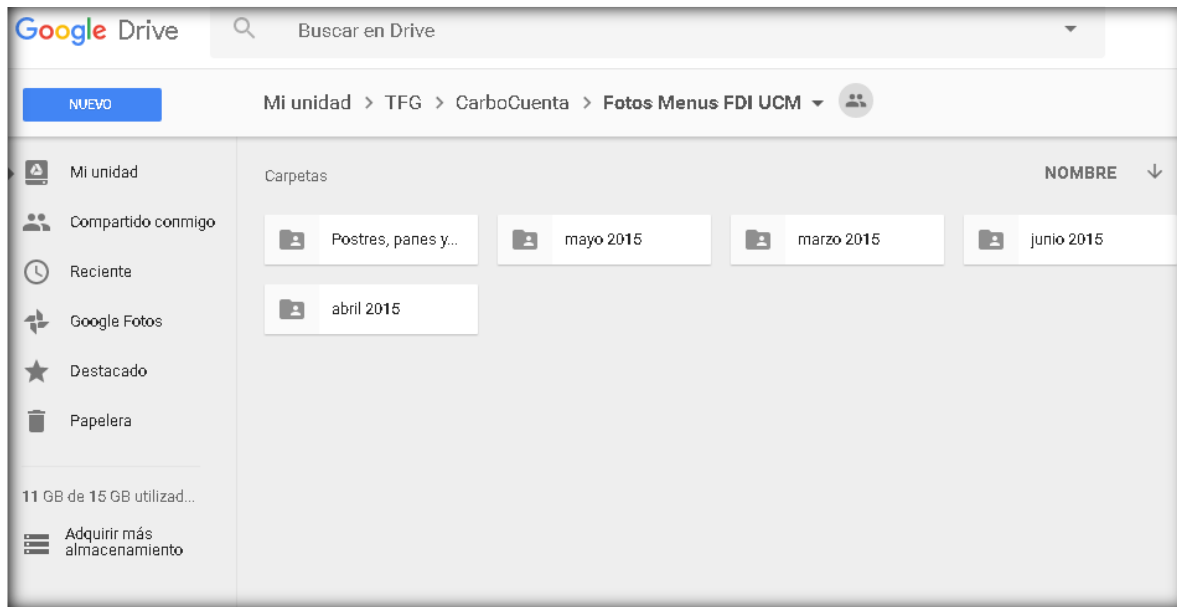


Ilustración 4: Almacenamiento por meses en Google Drive de las imágenes que compondrán la base del conocimiento

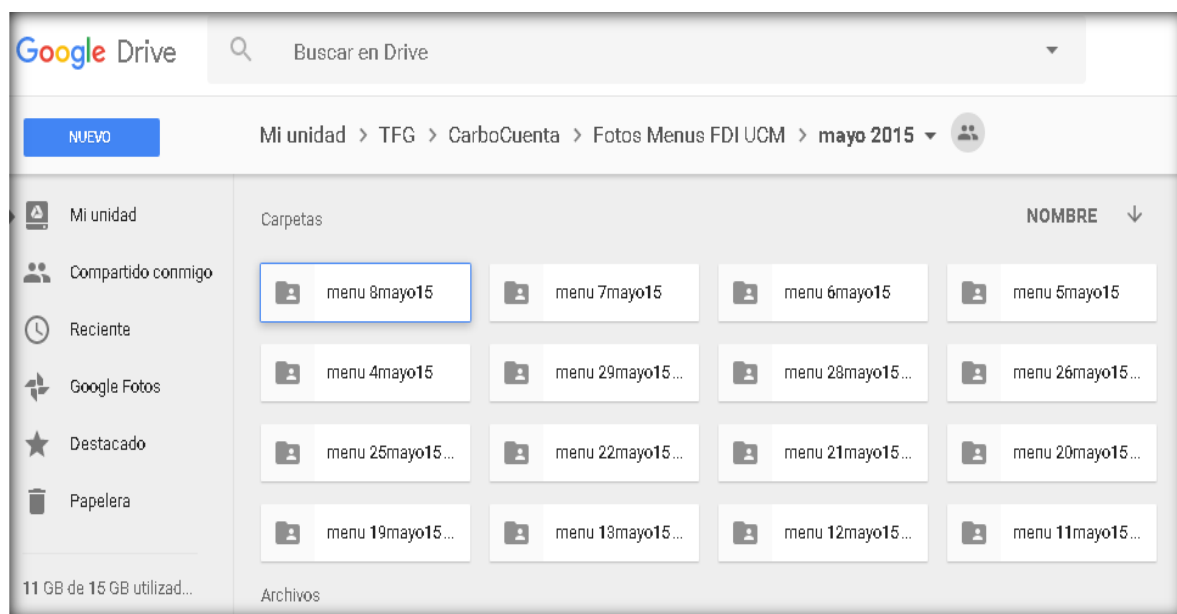


Ilustración 5: Diferentes menús diarios fotografiados.

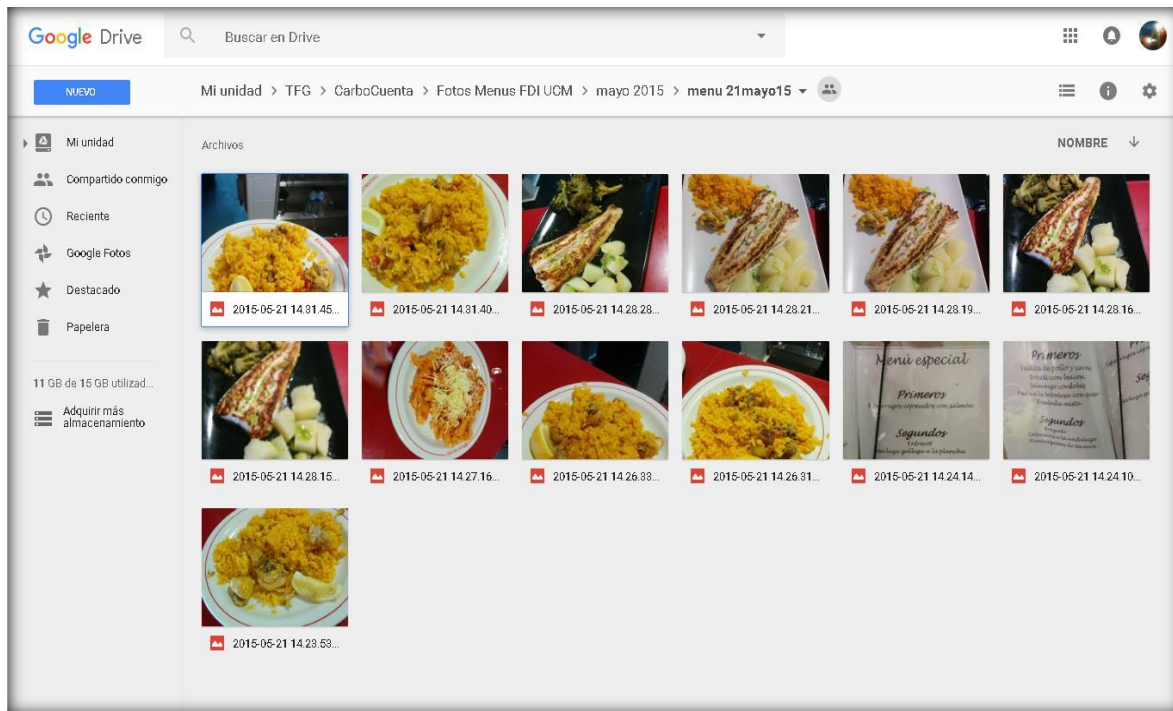


Ilustración 6: Ejemplo de Menús diarios para el 21 de mayo del 2015

Fase 1.B.

Tecnologías usadas: IDE MatlabR2015b, Open Office 4.1.1, Sublime Text 2.

Lenguaje de programación: Matlab.

Descripción algorítmica: Este script importa y usa librerías propias de Matlab para el tratado de imágenes aplicando máscaras de bits que la filtran la y guardaran en matrices.

Dicho script se aplicó a 50 muestras obteniendo así los pixeles que conforman la imagen. Estos pixeles están formados por 3 valores que corresponden a la cantidad de color rojo, verde y azul que describen cualquier imagen digital.

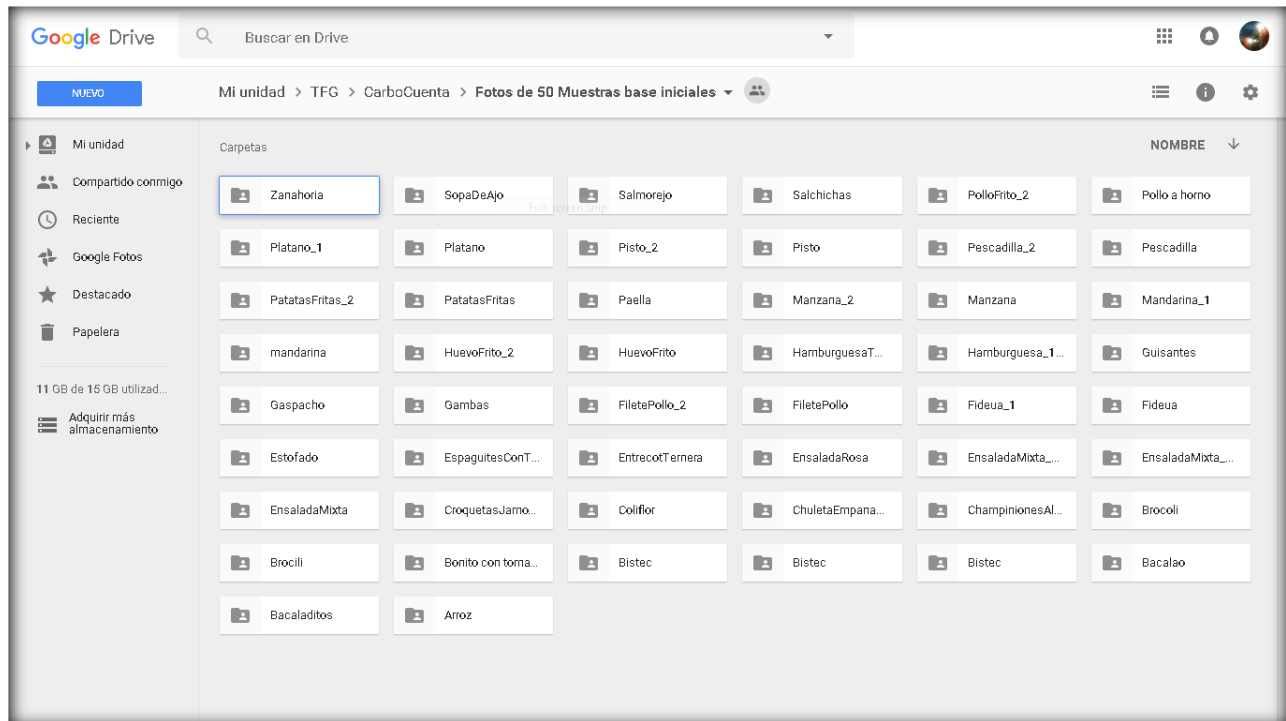


Ilustración 7: 50 muestras iniciales

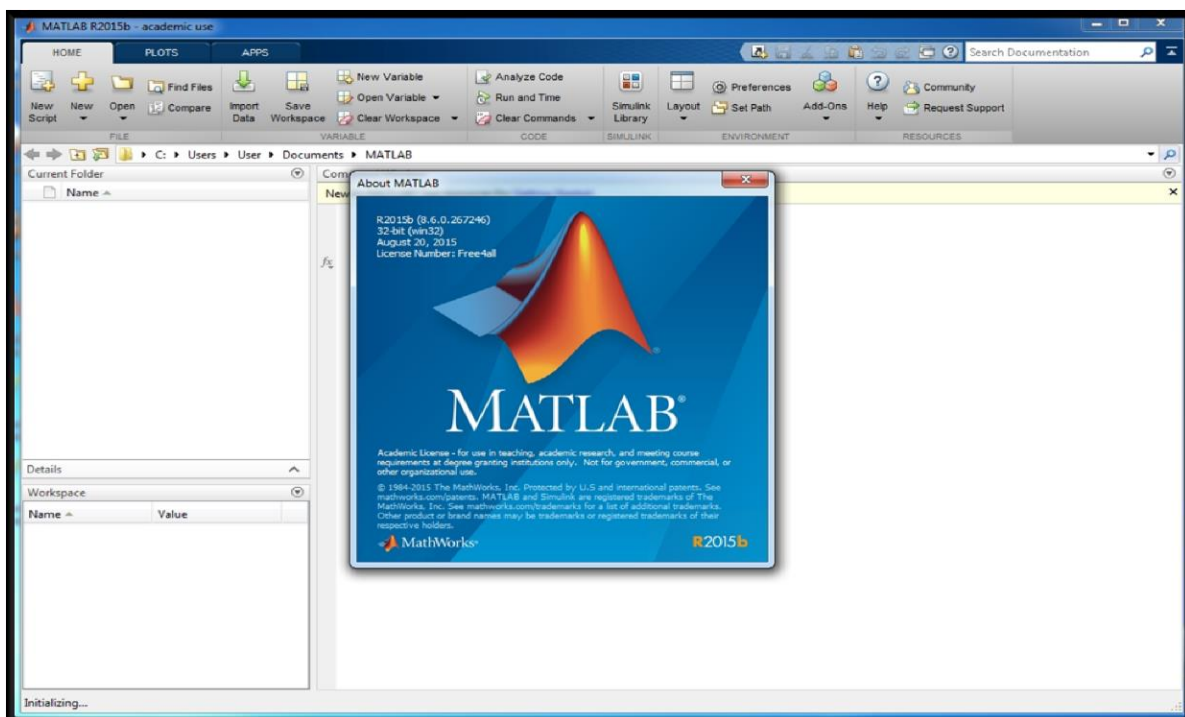


Ilustración 8: IDE MatlabR2015b

El script es capaz de aplicar filtros a la región que se ha seleccionado para analizar y luego extraer los pixeles de dicha región:

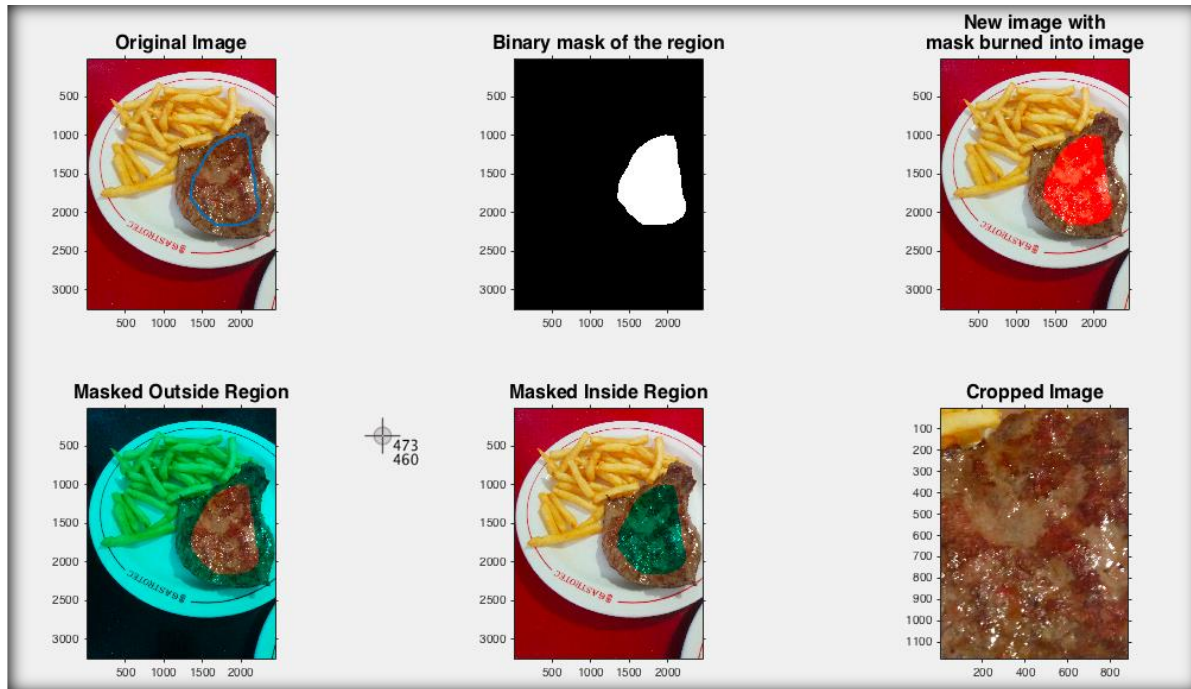


Ilustración 9: Filtros aplicados a la imagen para extraer los pixeles del área seleccionada

Generando finalmente un archivo .csv con todos los pixeles y un archivo.bmp con la región que se seleccionó:

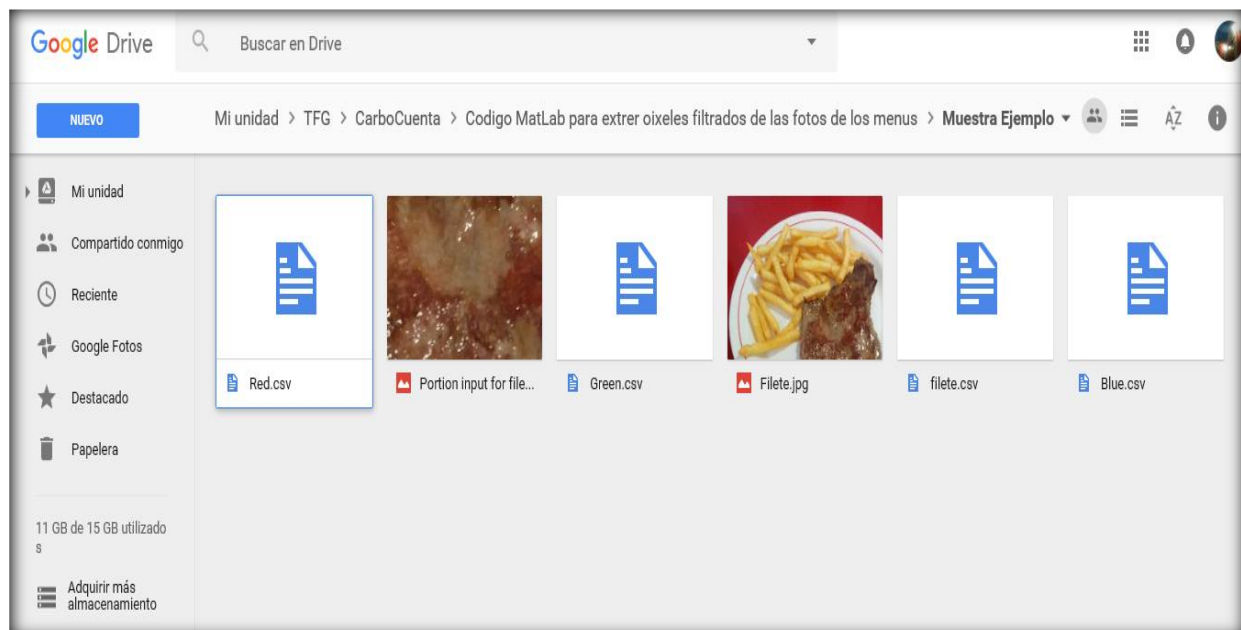


Ilustración 10: Archivos generados de la región seleccionada de la imagen muestra

En nuestro caso, dependiendo del tamaño de la región a tratar, podíamos obtener un archivo de salida de 100.000 hasta 1 millón de líneas de píxeles RGB, escogimos muestras de 300.000 píxeles ya que la potencia de los ordenadores requerida para archivos más densos era notable.

Fase 2. Entrenamiento de los algoritmos de IA

Tecnología usadas: IDE Eclipse versión luna reléase 2, Adminer 4.2.2, PHPMyAdmin 4.4.14.

Lenguaje de programación: Java 8, SQL, Bash Shell scripts.

Descripción algorítmica: Se hace uso de 6 clases Java:

- Bayes.java: Aplica las fórmulas de IA.
- DatabaseInsertion.java: Usa la librería mysql connector para insertar los resultados obtenidos en la BBDD.
- Determinant.java: Calcula el determinante de una matriz 3x3.
- Inverse.java: Calcula la inversa de una matriz 3x3
- Main.java: Ejecuta el programa
- ExecuteShellComand.Java: Lanza comandos bajo Shell script para realizar un backup de la BBDD.

Para esta segunda fase se creó un programa en Java que toma como entrada los píxeles previamente procesados por el script en Matlab, y da como salida estructuras que identifican a cada alimento en modo Bayesiano.[5].

En detalle se generan:

- Vector M:** Tiene la media de los valores RGB de todos los píxeles del alimento.
- Matriz C:** Matriz de 3x3 que identifica el alimento mediante una matriz de valores calculado con las fórmulas de IA mostradas a continuación
- **Inverse C:** La matriz inversa de C.

Función a estimar por máxima verosimilitud

Distancia de Mahalanobis

$$f(\mathbf{x}_i, \mathbf{w}) = p(\mathbf{x}_i / \mathbf{m}, C) = \frac{1}{(2\pi)^{d/2} |C|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_i - \mathbf{m})^t C^{-1} (\mathbf{x}_i - \mathbf{m}) \right\}$$

donde $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2\}$ es el vector de parámetros a "aprender" (estimar)

Resultado:

$$\hat{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad C = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{m}})(\mathbf{x}_i - \hat{\mathbf{m}})^t$$

$\mathbf{w} = \{\mathbf{m}, C\} \Rightarrow$ vector de parámetros aprendido

Ilustración 11: Formula de la distancia de Mahalanobis usada para calcular el grado de semejanza entre diferentes muestras

Cada clase sigue una distribución cuya densidad de probabilidad es:

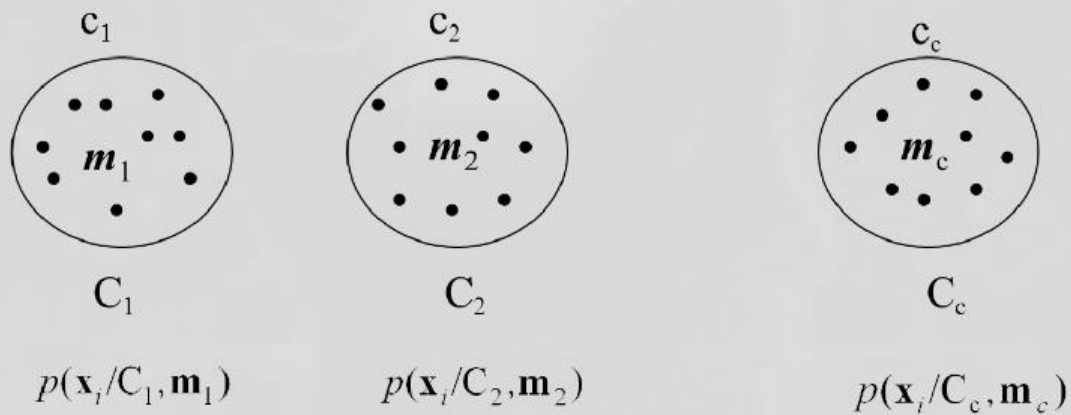


Ilustración 12: Clases Bayes de IA

Estadísticamente las ecuaciones Bayesianas:

$$\hat{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad C = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{m}})(\mathbf{x}_i - \hat{\mathbf{m}})^t$$

Son la media (\mathbf{m}) y la covarianza ($C = \sigma^2$), en una distribución de probabilidad Gaussiana o Normal, donde la estima de máxima verosimilitud para el vector media \mathbf{M} es la media simple y La estima de máxima verosimilitud para la matriz de covarianza es la media aritmética de las n matrices $(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^t$. [14]

Aplicando estas fórmulas obtenemos los datos necesarios para identificar cada alimento de forma matricial:

```

Main [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java (Dec 12, 2015, 2:54:33 PM)
10.084084084084083 15.409409409409406 10.323323323323322
10.084084084084083 15.409409409409406 10.323323323323322
10.084084084084083 15.409409409409406 10.323323323323322
12.084084084084083 17.409409409409406 12.323323323323322
9.084084084084083 14.409409409409406 9.323323323323322
8.084084084084083 13.409409409409406 8.323323323323322
9.084084084084083 14.409409409409406 9.323323323323322
10.084084084084083 15.409409409409406 10.323323323323322
12.084084084084083 15.409409409409406 10.323323323323322

Print vector Xi - M Trasposte:
7.084084084084083 7.084084084084083 11.084084084084083 10.084084084084083 12.084084084084083 11.084084084084083
5.409409409409406 5.409409409409406 8.409409409409406 9.409409409409406 8.409409409409406 10.409409409409406
-6.676676676676678 -6.676676676676678 -3.676676676676678 -3.676676676676678 -4.676676676676678 -2.676676676676678

Print Vector C:
[452.2872241610976, 591.9035131227322, 607.421262102943]
[591.9035131227322, 919.9615130646158, 1037.0297905513141]
[607.421262102943, 1037.0297905513141, 1256.9234900566232]

The inverse of filete is:
0.03248777709144293 -0.04581095938127149 0.022096462726669416
-0.045810959381271506 0.08013654569584054 -0.0439783605527609
0.02209646272666941 -0.043978360552760894 0.026401773073618573

The determinant of the Matrix filete is:
2489873.314519644
Connecting to database...
  
```

Ilustración 13: IDE Eclipse generando la data IA

Foods_IA

Contenido

Mostrar estructura

Modificar tabla

Nuevo Registro

Condición

Ordenar

Limite

Longitud de texto

Acción

50

100

Mostrar

`foods_ia` LIMIT 50 (0.002 s)

Modificar

id	name	vectorMred	vectorMgreen	vectorMblue	inverseC00	inverseC01
0	Filete	130.67232767232767	90.00599400599401	63.92107892107892	0.02093943552227767	-0.022953607204835837
1	Arroz	197.02102102102103	216.46946946946946	205.63863863863864	0.06138019930096029	-0.0640072424214722
2	Filete	126.92692692692692	81.2032032032032	53.9029029029029	0.018702593918909723	-0.017422537029558604
3	Patatas fritas	203.36136136136136	163.52752752752752	90.36136136136136	0.04111264344720158	-0.020351630177708913
4	Bacaladitos	121.03180906487171	96.15684837204559	52.39320772035389	0.0022591974026571334	-0.0028958080011328158
5	Bacalao	142.05724878898513	109.22349189872814	53.269915129532855	0.005950446590365582	-0.008299595071053562
6	Bistec	96.63303688430138	79.02776648960628	62.243622987195316	0.03266015867109689	-0.03907504577516534
7	Bistec	133.98112631063984	97.01141242675902	72.39635758184235	0.01786833012358976	-0.01949127324391116
8	Bistec	119.78454182070047	79.10765259430485	54.18908165715806	0.015078342552258404	-0.014776399925207953
9	Bonito con tomate	143.0608243345762	59.8017206592094	11.553699982244606	0.004533170338983176	-0.005963918841775992

Ilustración 14: BD generada con IA data de cada alimento

Fase 3. Generar la base de datos con la información nutricional para cada alimento y su salvado en el servidor API

Tecnologías usadas: Apache server 2.4.18, MySQL 5.6.27, Sequel pro 1.0.2, Adminer 4.4.2, PhpmyAdmin 4.4.14, XAMPP 5.6.3

Lenguajes de programación: SQL, PHP 5.6.14.

Descripción algorítmica: Haciendo uso de comandos Bash Shell de importación y manejo de BBDD se realizó la mayoría de las operaciones de llenado de la BD.

Para ello se hicieron uso de dos bases de datos con diferentes tipos de información que fueron tomadas como entradas:

- Estática: Del CESNID y con la información nutricional de más de 700 alimentos.
- Dinámica: La API de USDA (United States Department of Agriculture) Food Composition Databases <https://ndb.nal.usda.gov/ndb/search>. A la cual se le pueden hacer consultas web sobre cualquier alimento y esta devolverá los valores nutricionales de alimento consultado, para esto tuvimos que hacer uso de la API de Google translator tanto para traducir las respuestas como las peticiones.

Generando así una BB con más de 40 valores nutricionales para cada alimento, que van desde carbohidratos, calorías, grasas, azúcares hasta proteínas, vitaminas, minerales, ácidos, etc. En la siguiente ilustración se pueden observar todos:

Column	Type	Comment
id	int(111) Auto Increment	
Alimento	varchar(222) NULL	
Cantidad recomendada (g)	varchar(111) NULL	
Hidratos Carbono (g) (no contada la fibra)	varchar(111) NULL	
Calorías (Kcal)	varchar(111) NULL	
Proteína Total (g)	varchar(111) NULL	
Grasa Total (g)	varchar(111) NULL	
Colesterol (mg)	varchar(111) NULL	
Azúcares Totales (g)	varchar(111) NULL	
Proteína Animal (g)	varchar(111) NULL	
Proteína Vegetal (g)	varchar(111) NULL	
Porción Comestible	varchar(111) NULL	
Agua (g)	varchar(111) NULL	
Polisacáridos Totales (g) (no contada la fibra)	varchar(111) NULL	
Fibra Alimentaria (g)	varchar(111) NULL	
Ácidos Grasos Saturados (g)	varchar(111) NULL	
Ácidos Grasos Mono (g)	varchar(111) NULL	
Ácidos Grasos Polisaturados (g)	varchar(111) NULL	
Sodio (Na) (mg)	varchar(111) NULL	
Calcio (Ca) (mg)	varchar(111) NULL	
Potasio (K) (mg)	varchar(111) NULL	
Fósforo (P) (mg)	varchar(111) NULL	
Hierro (Fe) (mg)	varchar(111) NULL	
Magnesio (mg)	varchar(111) NULL	
Cinc (mg)	varchar(111) NULL	
Carotenos (µg)	varchar(111) NULL	
Etanol (g)	varchar(111) NULL	
Retinol (µg)	varchar(111) NULL	
Vitamina A: Total (µg)	varchar(111) NULL	
B1: Tiamina (mg)	varchar(111) NULL	
B2: Riboflavina (mg)	varchar(111) NULL	
B3: Niacina (mg)	varchar(111) NULL	
B6: Piridoxina (mg)	varchar(111) NULL	
B9(M): Ácido Fólico (µg)	varchar(111) NULL	
B12: Cobalamina (µg)	varchar(111) NULL	
C: Ácido ascórbico (mg)	varchar(111) NULL	
D: Calciferol (µg)	varchar(111) NULL	
E: Tocoferol (mg)	varchar(111) NULL	
Grupo	varchar(111) NULL	
Subgrupo	varchar(111) NULL	
More Info	varchar(222) NULL	

Ilustración 15 Valores nutricional de los alimentos en la BD

127.0.0.1/adminer?username=root&db=tf&select=CESNID_Alimentos_Composicion

Language: English MySQL » Server » tf » Select: CESNID_Alimentos_Composicion

Adminer 4.2.3 4.2.5

DB: tf

SQL command Import Export Create table

select CESNID_Alimentos_Composicion
select Foods_IA
select User_Data

Select data Show structure Alter table New item

Select Search Sort Limit 50 Text length 100 Action

SELECT * FROM `CESNID_Alimentos_Composicion` LIMIT 50 (0.051 s) Edit

	Modify	id	Alimento	Cantidad recomendada (g)	Hidratos Carbono (g) (no contada la fibra)	Calorias (Kcal)	Proteina Total (g)	Grasa Total (g)
<input type="checkbox"/>	edit	1	Aceite de cacahuete	100	0	895.2929077148	0	99.477
<input type="checkbox"/>	edit	2	Aceite de colza	100	0	899.1000366211	0	99.9
<input type="checkbox"/>	edit	3	Aceite de girasol	100	0	899.1000366211	0	99.9
<input type="checkbox"/>	edit	4	Aceite de grano de uva	100	0	899.05859375	0	99.8954
<input type="checkbox"/>	edit	5	Aceite de hígado de bacalao	100	0	896.234375	0	99.5816
<input type="checkbox"/>	edit	6	Aceite de maíz	100	0	899.1000366211	0	99.9
<input type="checkbox"/>	edit	7	Aceite de nuez	100	0	898.1171875	0	99.7908
<input type="checkbox"/>	edit	8	Aceite de oliva	100	0	899.1000366211	0	99.9
<input type="checkbox"/>	edit	9	Aceite de oliva virgen	100	0	899.1000366211	0	99.9
<input type="checkbox"/>	edit	10	Aceite de soja	100	0	899.1000366211	0	99.9
<input type="checkbox"/>	edit	11	Mantequilla	100	0	891	0	99
<input type="checkbox"/>	edit	12	Margarina 3/4 vegetal	100	0.21	752.8649535775	0.69	83.2517
<input type="checkbox"/>	edit	13	Margarina light	100	0.5	543.860039115	0.33	60.06
<input type="checkbox"/>	edit	14	Margarina mixta	100	0.4	373.7500001192	1.35	40.75
<input type="checkbox"/>	edit	15	Margarina vegetal enriquecida	100	0.952381	739.0094874501	0.2	81.6
<input type="checkbox"/>	edit	16	Aperitivo de maíz: Generico tipo nachos doritos fritos + similares	100	0.4	731.0499634743	0.45	80.85
<input type="checkbox"/>	edit	17	Galletas saladas	100	57.3	533.9833126068	6.83333	30.05
<input type="checkbox"/>	edit	18	Galletas saladas con queso	100	49	458.4	10	24
<input type="checkbox"/>	edit	19	Palomitas de maíz con aceite con sal	100	49.1	486.0970985413	11.8	26.2552
<input type="checkbox"/>	edit	20	Palomitas de maíz sin aceite sin sal	100	47.2	532.5000152588	9	31.9667
<input type="checkbox"/>	edit	21	Patatas chips	100	62.9	367.4000053406	12	4.2
<input type="checkbox"/>	edit	22		100	49.4667	545.5575866699	6.5	34.8545
<input type="checkbox"/>	edit			100	57.0991	474.5462867737	7.05	22.25

127.0.0.1/adminer?username=root&db=tf&sql=

Ilustración 16: Información nutricional de los alimentos en la BD

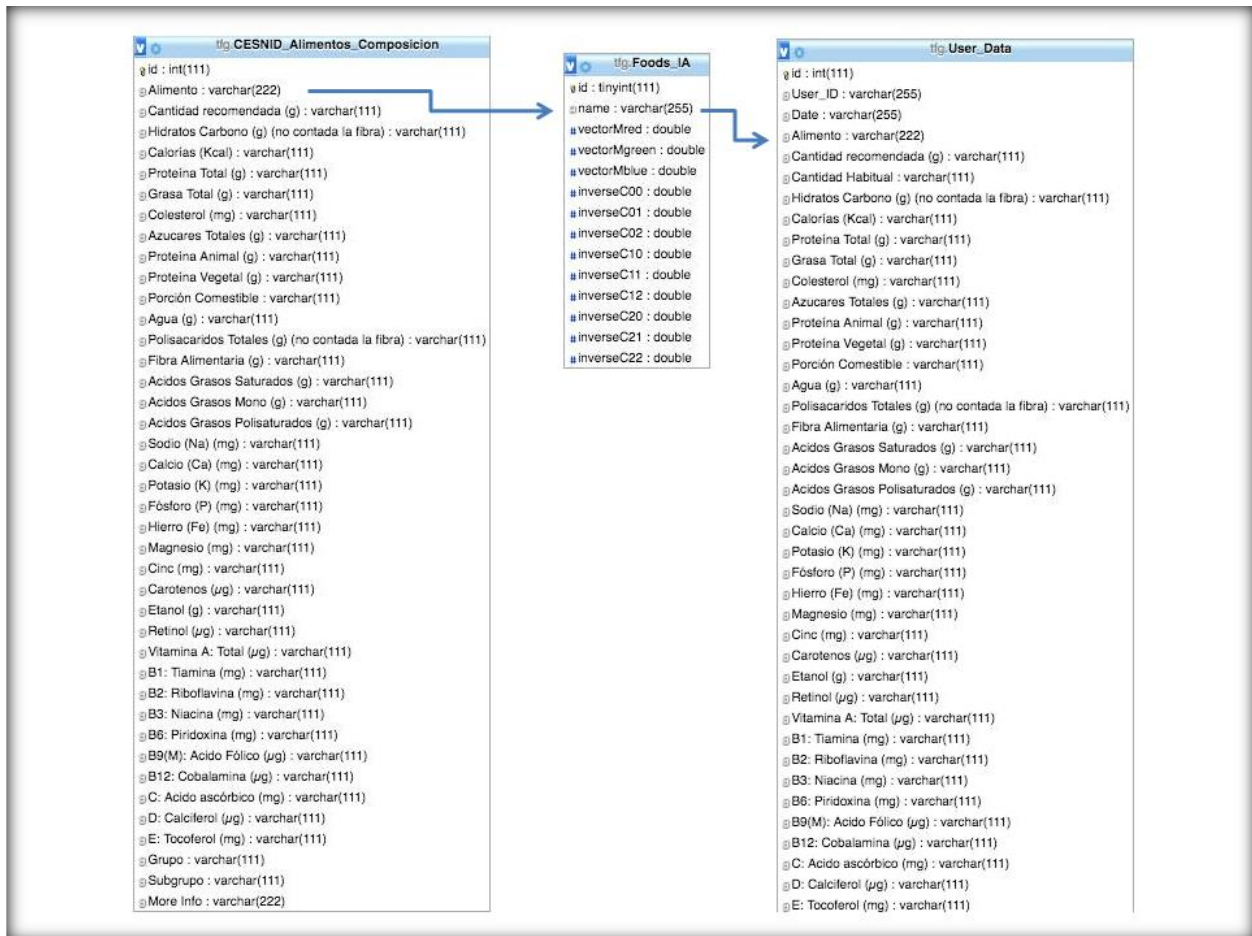


Ilustración 17: Esquema y detalle de la BD

Fase 4. Creación del servidor de aplicaciones (API) y el cliente (Android App)

Fase 4.A. Creación del Servidor API

Tecnologías usadas: IDE Netbeans 8.0.2, API Google Translator V2, API de USDA (departamento de agricultura de los Estados Unidos) release 28,

Lenguajes de programación: PHP 5.6.14, Bash Shell script, JSON.

Descripción algorítmica: Para el desarrollo de esta fase se estructuró el código haciendo uso de los siguientes **patrones de Ingeniería del Software**:

- **MVC (Modelo Vista Controlador)**

-Modelo: Los modelos son las representaciones que construimos basadas en la información con la que operará el sistema.

-Vista: Las vistas son los elementos que permiten que un usuario interactúe con el sistema. Éstas están desarrolladas mediante el lenguaje JSON.

-Controlador: Son aquellas clases que nos permiten dar vida a las vistas de nuestra aplicación, permitiendo obtener y proveer la información que el usuario emplea. Estas clases son desarrolladas mediante el lenguaje de programación Java.

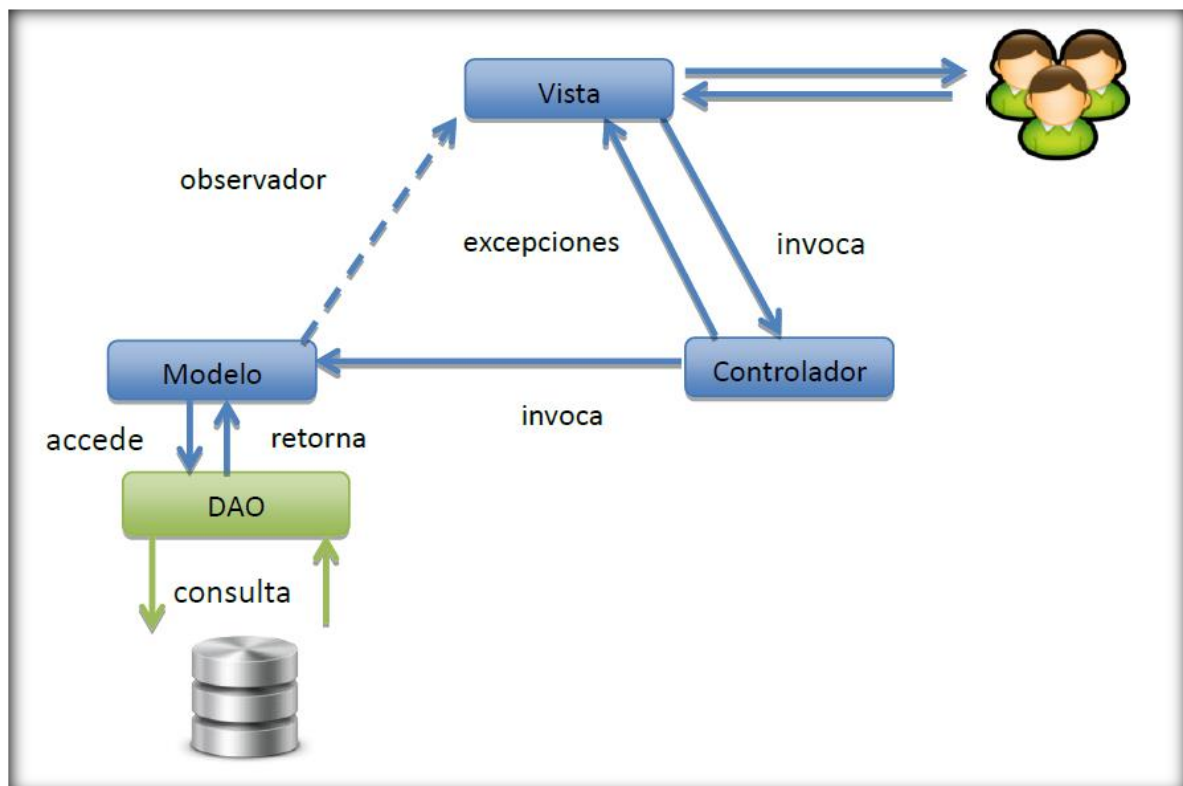


Ilustración 18: Esquema general del patrón MVC

- **Patrón DAO:** Suministra una interfaz común entre la app y la BBDD. Y filtra toda las peticiones para evitar Inyección de SQL.
- **Patrón Factory:** Utiliza una clase constructora dedicada a la construcción de objetos de un subtipo de un tipo determinado.
- **Patrón Dispatcher:** Lanza la acción recibida por el controlador mediante petición HTTP.
- **Patrón Singleton:** Garantiza que una clase sólo tenga una instancia, proporcionando un punto de acceso global a dicha instancia.

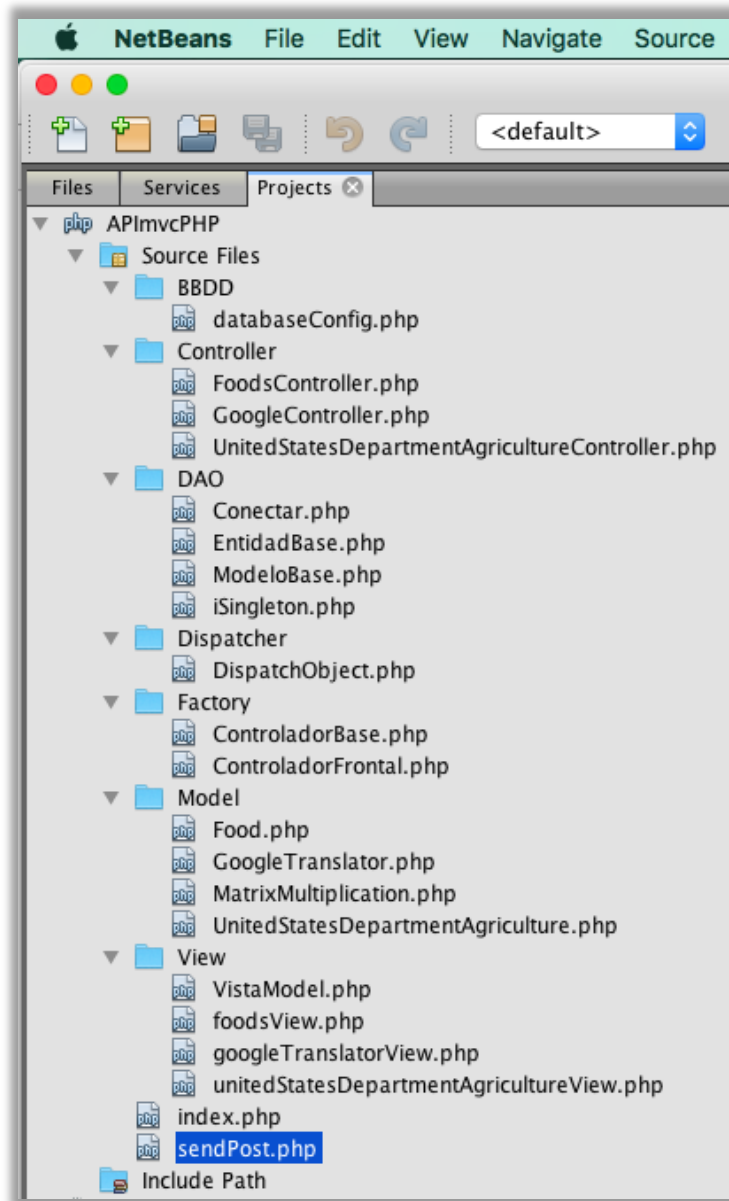


Ilustración 19: Patrones IS usados en el servidor API

Fase 4.B.1. Creación del Cliente (APP Android)

Tecnologías usadas: IDE Android Studio 2.1, GitHub 2.1.6, Hilos multi-concurrencia, Filesilla FTP 3.15 server.

Lenguajes de programación: Java 8, XML, SQL, JSON.

Descripción algorítmica: La App está enfocada de tal modo que use el mínimo de recursos y deje más bien al servidor realizar la mayor parte de las operaciones y cálculos, de esta forma el dispositivo móvil no tiene que ser muy potentes además de que se optimiza el uso de la batería..



Ilustración 20: Arquitectura del proyecto

En la ilustración se puede observar la arquitectura empleada: Un servidor para este proyecto el cual trabaja como API para iPhone y Android. Este servidor es el encargado de recibir los pixeles desde la App y devolver los posibles alimentos coincidentes, para ello se emplean dos métodos:

- 1- La comparación de las distancias entre la medias RGB de los pixeles de cada alimento con la media RGB de los pixeles de la fotografía actual.

- 2- La comparación entre las distancia de Mahalanobis Bayesianas entre los pixeles de los alimentos de la BD y los recibidos por la fotografía actual.

En ambos dos métodos, una diferencia grande indicara una no similitud y un resultado cercano a cero indicara una muy probable similitud entre la foto de alimento y el alimento analizado en la BD.. Por ejemplo en el caso más sencillo, el caso 1, si en el servidor se tiene que un tomate tiene de media los colores RGB=255,50,0 (mucho R=rojo=255 y poco G=Green=Verde=50 y nada de B=Blue=Azul) y se le envía una muestra con la media de los valores RGB (250,40,0) la resta de las distancias dará valores cercanos cero por lo que se concluye el alimento es parecido.

En un futuro también se configurará un servidor de apoyo y backup, y se hará uso de la BBDD del proyecto de gIUCModel para dar al usuario respuestas personalizadas y llevar un control de su dieta.

En el proyecto Android destacan los siguientes directorios:

- **app/src/main/java**, Contiene las clases que definen las actividades, modelos y utilidades del sistema.
- **app/src/main/res**, Es el directorio que contiene los recursos como son las imágenes, los XML de las vistas y las cadenas que aparecen en dichas vistas.
- **res/drawable**. Son los directorios que contienen las imágenes y recursos XML para la personalización de elementos de las vistas.
- **res/layout**, Directorio que contiene los archivos XML de las vistas.
- **res/values**. Directorio que contiene los XML con las cadenas que muestran las vistas, los recursos de colores y la definición de estilos de las pantallas.

Fase 4.B.2. Parametrización del software base para Android

Las aplicaciones Android poseen un fichero llamado AndroidManifest.xml, en el que se especifica todos los elementos, pantallas y permisos que la aplicación va a necesitar durante su ejecución.

En esta APP en específico tenemos:

1. **API mínima y API máxima.** Especifica cuales son las versiones del SDK mínima y máxima que emplea la aplicación.

```
< uses-sdk android:minSdkVersion="4" android:targetSdkVersion="23"/>
```

2. **Permisos para uso del internet y la camera.**

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
  android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-feature android:name="android.hardware.camera" />
```

3. **Actividades de la aplicación.** (Son las diferentes pantallas que posee el sistema).

```
<!-- Main Activity -->
<activity android:name=".MainPhotoIntentActivity"
  android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>

<!-- All food Activity -->
<activity
  android:name=".AllFoodsActivity"
  android:label="All foods" >
</activity>

<!-- View food Activity -->
<activity
  android:name=".ViewFoodActivity"
  android:label="Food nutrients composition" >
</activity>
```

```

<!-- Add food Activity -->
<activity
  android:name=".AddFoodActivity"
  android:label="Add food to DDBB" >
</activity>

```

Fase 4.B.3. Construcción del sistema

Este punto es de vital importancia, pues presenta la herramienta que permite automatizar la construcción de nuestro sistema.

La herramienta seleccionada para este fin es Gradle, pues es una herramienta que permite la automatización de la compilación del proyecto, importación de bibliotecas a emplear en el mismo, automatización de sistema de pruebas y despliegue del sistema.

Para entender mejor lo simple que puede llegar a ser, se va a exponer un ejemplo del esqueleto del fichero “build.gradle” empleado por Android Studio para esta labor.

```

apply plugin: 'com.android.application'
android {
  compileSdkVersion 21
  buildToolsVersion "23.0.2"

  defaultConfig {
    applicationId "com.example.android.photobyintent"
    minSdkVersion 4
    targetSdkVersion 23
  }
}

// Top-level build file where you can add configuration options common to all sub-
// projects/modules.
buildscript {
  repositories {
    jcenter()
  }
  dependencies {
    classpath 'com.android.tools.build:gradle:2.1.0'
  }
}

```

Hay **4 áreas** principales en este fichero:

1. Plugin 'com.android.application'.

Este plugin proporciona todo lo necesario para construir y probar aplicaciones

2. buildscript {...}.

En esta parte del Gradle es donde se configura el código. En este caso, se declara que se utiliza el repositorio de JCenter, y que hay una dependencia en el classpath de un artefacto de JCenter. Dicho artefacto es la biblioteca que contiene el plugin de Gradle para Android en la versión 2.1.0.

3. dependencies{...}

Es el manejador de bibliotecas internas y externas. Permite simplificar la inclusión de bibliotecas en el proyecto.

4. . Android{...}.

Es el encargado de configurar los parámetros para la construcción de Android. Por defecto, solo es necesario que incluya la propiedad `compileSdkVersion` para poder compilar. En este caso se ha compilado con la versión 21.

Fase 4.B.4. Concurrency, paralelismo e Hilos

Útiles para aprovechar los actuales dispositivos con múltiples procesadores. Éstos permiten que varias tareas o cálculos se lleven a cabo en paralelo aumentando el rendimiento y sensación de rapidez y fluidez de la App.

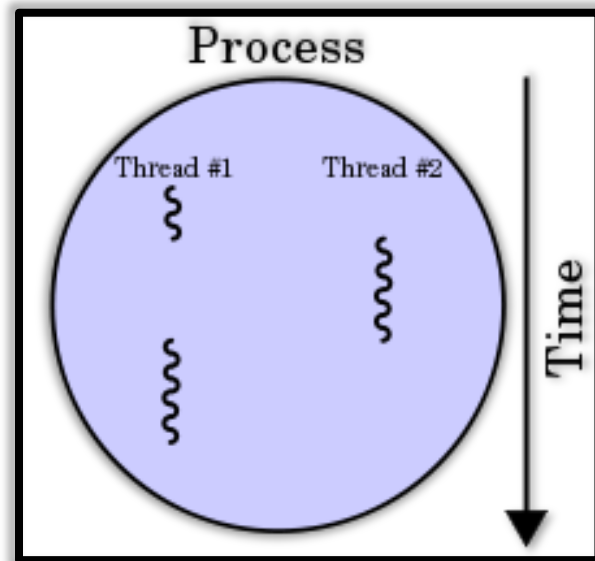


Ilustración 21: Definición gráfica de los hilos multi tarea en el SO

Fuente: [https://en.wikipedia.org/wiki/Thread_\(computing\)](https://en.wikipedia.org/wiki/Thread_(computing))

En concreto las siguientes llamadas en esta App hacen uso de hilos de multi concurrencia:

- Cada vez que se realiza una petición de lectura./escritura al servidor mediante peticiones HTTP POST & GET.
- En el procesamiento de los pixeles capturados en la fotografía
- Cada vez que se guarda la información del usuario
- Cuando se llaman a las APIS externas
- Cuando cargamos los valores de los alimentos.

Un ejemplo de uso se puede apreciar en la siguiente imagen en la cual la clase que implementa los hilos es *AsyncTask*:

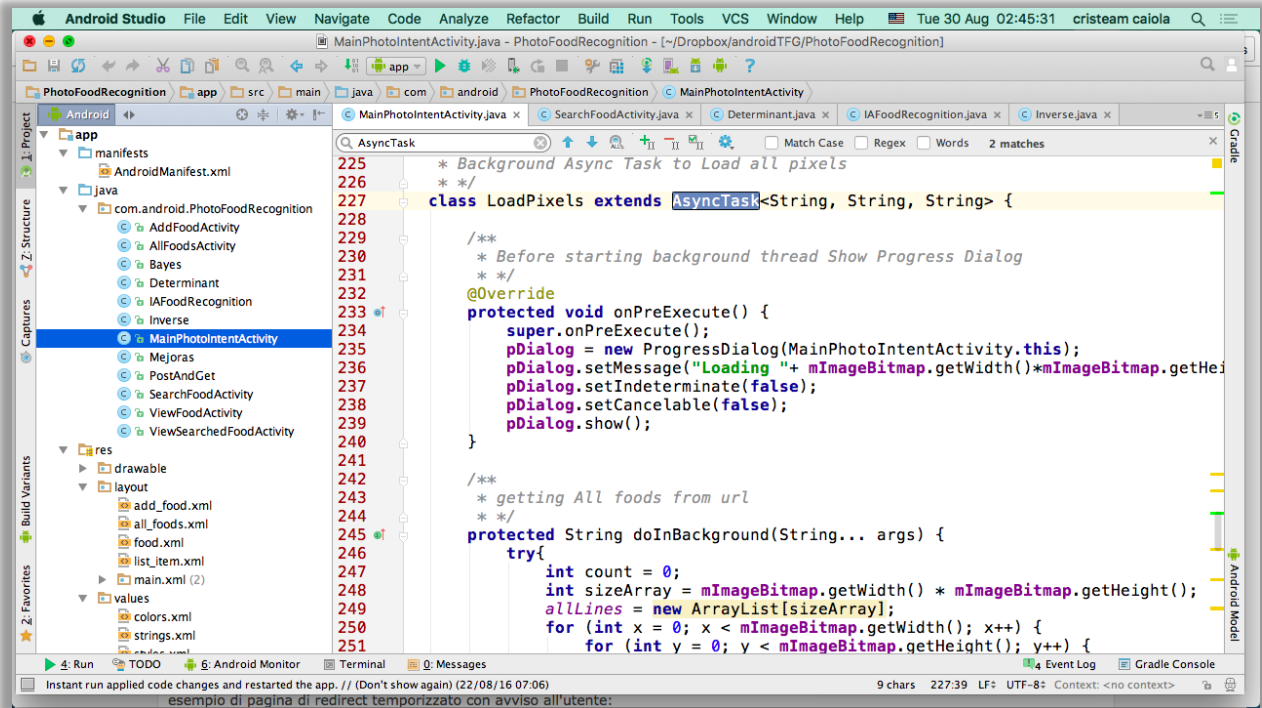


Ilustración 22: Ejemplo de la implementación de hilos en Android

WEB DEMO

Una Demo completa de todas estas fases está disponible en:

<http://carbocuenta.dacya.ucm.es/demo.php>

Lenguajes de programación usados: HTML 5, CCS 3, Javascript 1.7, JQuery 2.1, Bootstrap 3.1.0

Descripción algorítmica: En esta demo se hace un resumen de las 4 fases anteriores para ello se programó la web importando la librería de Bootstrap para generar las barras de progreso y jQuery para dar estilos y comportamiento a los botones. HTML5 y CSS3 fueron empleados para generar la estructura de la página.

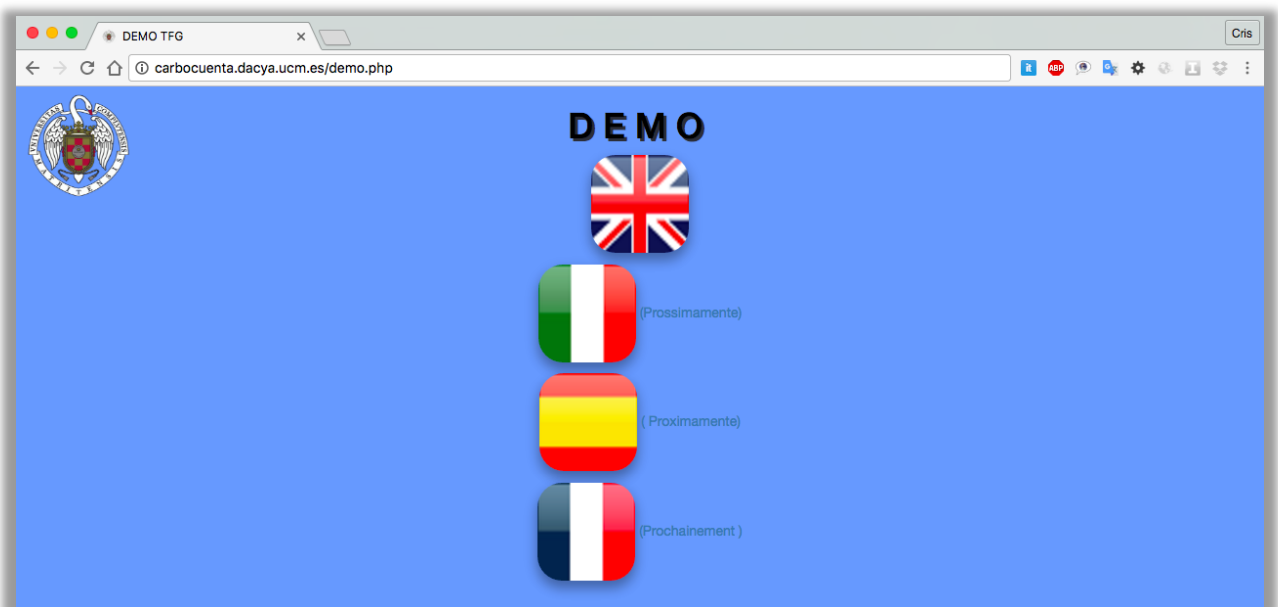


Ilustración 23: Demo WEB

Capítulo 4. Análisis de Ingeniería del Software aplicados al proyecto

4.1. Análisis de rendimiento de la App

Puesto que la Ingeniería del Software define entre sus requisitos no funcionales el del análisis del rendimiento, a continuación se ha hecho mediante dos aplicaciones monitoras diferentes, un análisis de uso de RAM y porcentaje de uso de CPU antes y después de iniciar al APP.

Para ello, vamos a observar las gráficas de consumo de CPU y RAM. Para entender las gráficas se dan algunas indicaciones:

En el primer círculo, el color azul indica el consumo de la aplicación junto otras aplicaciones en un estado normal, el color violeta es el consumo de las aplicaciones del sistema y el color amarillo es la CPU en desuso.

En cuanto a la gráfica de la RAM, es sencilla, el color amarillo indica la memoria consumida entre la aplicación y el resto de las aplicaciones en un estado normal, y el color violeta indica que está en desuso.

Resultados antes de ejecutar la App en NEXUS 5 con programa monitor A:

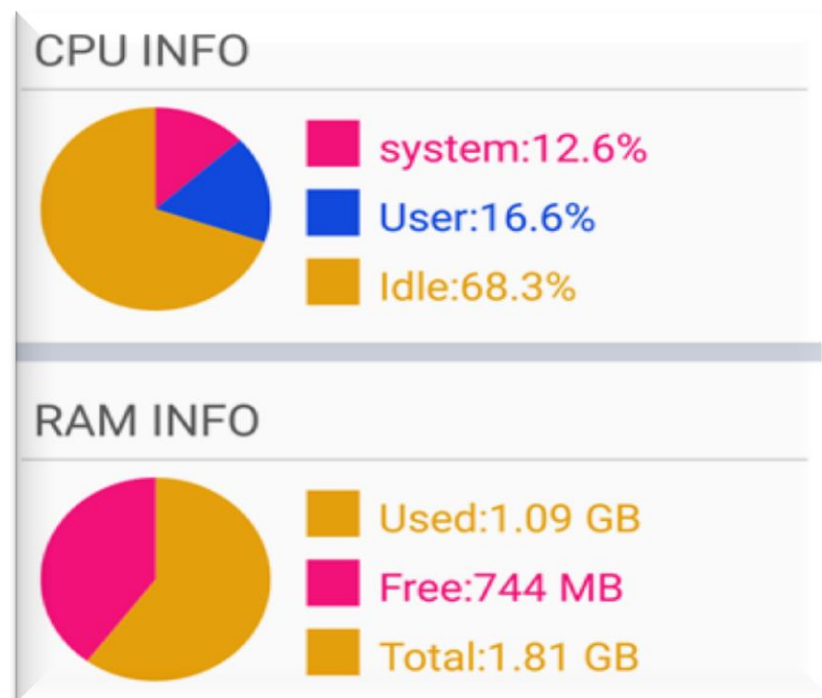


Ilustración 24: Rendimiento con programa monitor A antes de ejecutar App

Antes de ejecutar la App en NEXUS 5 con programa monitor B:

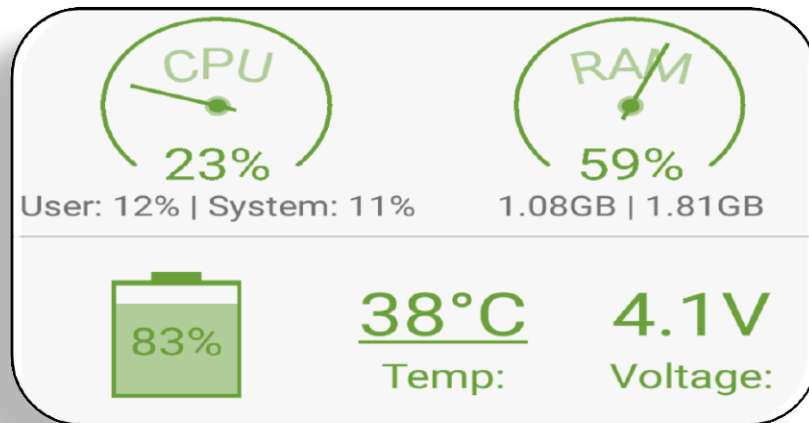


Ilustración 25: Rendimiento con programa monitor B antes de ejecutar App

Después de ejecutar la App con programa monitor A:

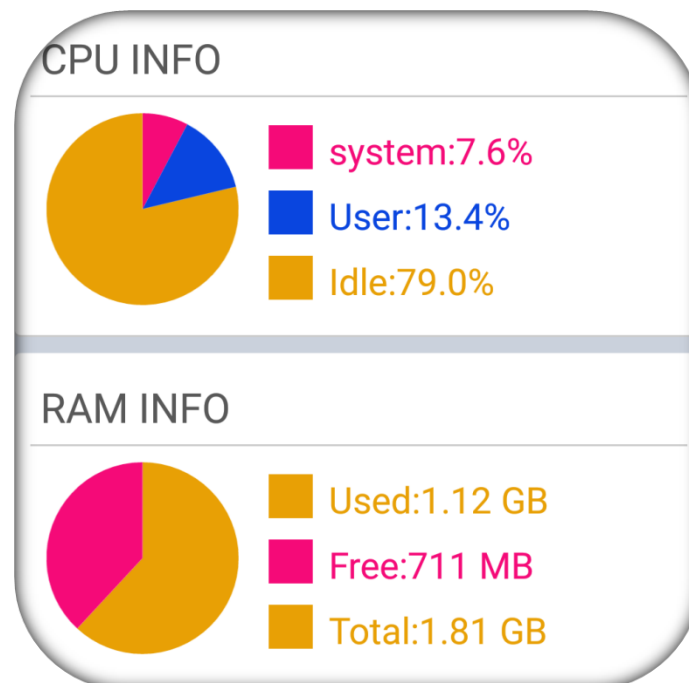


Ilustración 26: Rendimiento con programa monitor A despues de ejecución

Después de ejecutar la App con programa monitor B:

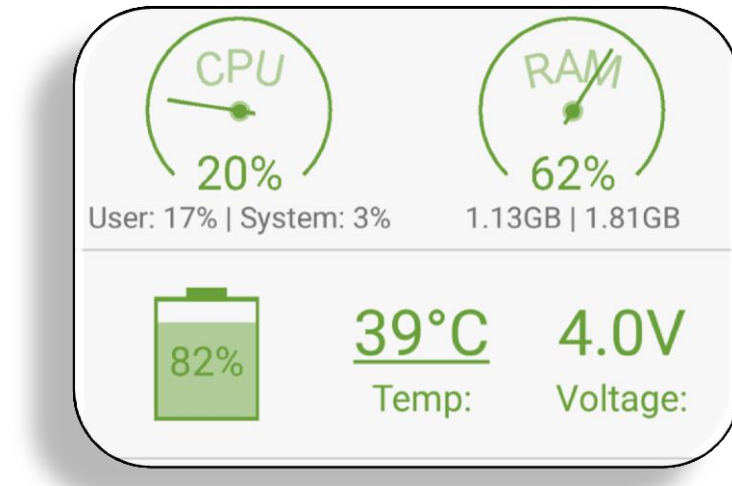


Ilustración 27: Rendimiento con programa monitor B después de ejecución

En conclusión como se puede ver en las imageens, el consumo y diferencia de CPU y memoria RAM entre el sistema y demás aplicaciones antes y después de lanzar la App es muy bajo, el CPU no varían en más de 10 unidades y la memoria nunca llega a llenarse, por lo que se da por hecho que la App no afecta al rendimiento del SO Android en un dispositivo de gama media.

4.2. Análisis de riesgos

Durante el desarrollo de cualquier proyecto informático existen una serie de riesgos que originan costes y/o retrasos en las entregas. A continuación se van a mostrar una serie de riesgos que existen en este proyecto.

1. Inexperiencia con la tecnología.

Puede darse la situación que la tecnología a emplear sea desconocida o no se tenga demasiado conocimiento de la misma, repercutiendo en los tiempos de entrega, ya sea por lentitud o por fallos.

2. Lentitud en la toma de decisiones.

En la etapa final del proyecto, al residir el autor en Londres y el director en Austria, los tiempos de respuestas entre correos y comunicación en general podrían portar retrasos.

3. Cambios en los requisitos.

Como en todo proyecto el cambio de requisitos siempre es habitual pero con buena comunicación su impacto es mínimo.

4. Cambios en las licencias, herramientas y APIs

Se supone que las licencias de uso de Android y de su IDE Android Studio son gratuitas bajo la licencia .GNU GPL*

Así como la API de Google translator siga siendo gratuita siempre y cuando no se superen ciertas tasas de uso. En la fecha de Agosto del 2015 era de más de 100.000 caracteres en 100 segundos o 2millones de caracteres por día.

Por ultimo también se da por hecho que la API del departamento de agricultura de los Estados Unidos USDA siga siendo gratuita.

*GNU GPL: Licencia que establece que todo software que posee dicha licencia es software libre y que cualquiera puede estudiar, compartir y modificar dicho software

5. Nuevas versiones de la API de Android.

Se puede dar la situación que Android actualice su SDK con nuevas funciones durante el desarrollo obligando a cambiar implementaciones ya hechas o a diseñar nuevas.

6. Ocupaciones extra laborales.

En este caso el autor ha tenido que compaginar el desarrollo de TFG con un trabajo a tiempo completo y otro a tiempo parcial de lunes a sábado..

7. Enfermedades.

Es algo normal que durante el desarrollo del proyecto se pueda llegar a estar enfermo, lo que podría provocar algún retraso.

Tabla de Análisis del impacto de riesgos contemplados.

Riesgo	Probabilidad	Impacto	Prioridad
Inexperiencia con la tecnología	media	medio	alta
Lentitud en la toma de decisiones	baja	medio	media
Cambios en los requisitos	media	alto	alta
Cambios en las licencias, herramientas y APIs	baja	alto	alta
Nuevas versiones de la API de Android	media	medio	media
Ocupaciones extra laborales	alta	alta	alta
Enfermedades	baja	medio	media

Tabla 1: Análisis de riesgos

Observando la tabla anterior, se puede concluir que es necesario tener en cuenta los tiempos que el trabajo permitirá compatibilizar para desarrollar la App aparte de conocer la tecnología a emplear, pues con ello se logrará acelerar el proyecto.

4.3. Análisis de Costes, fortalezas, oportunidades, debilidades y amenazas

4.3.1. Análisis de costes

En todo proyecto informático se debe de contemplar una serie de costes derivados de diferentes factores o herramientas empleadas para llevar a cabo el proyecto. Estos son: hardware, software y humano.

- **Hardware.** Ordenadores y Smartphone personales. Adquiridos antes de comenzar este proyecto.
- **Software.** El proyecto se desarrolló bajo herramientas de libre uso.
- **Humano:** Este proyecto se desarrolló como estudiante de la UCM y bajo acuerdo de confidencialidad con el director del proyecto

4.3.2. Fortalezas

Respecto a otras aplicaciones, actualmente en el mercado no hay muchas parecidas, para la fecha se encontraron sobre todo una aplicación similar desarrollada por la universidad de Tokio como tesis doctoral en vectores Fisher. Dicha app se encuentra en el repositorio de google drive: <https://drive.google.com/open?id=0B3vvnBgau2qdaFRXTzRPclloaTQ>.

4.3.3. Oportunidades

Al encontrarse este App bajo el cobijo del macro proyecto de glUCModel, tenemos un gran abanico de oportunidades ya que el proyecto está en continuo avance con aportaciones regulares.

4.3.4. Debilidades

La principal debilidad fue el tiempo que se le pudo dedicar al desarrollo como estudiante. A tiempo completo son muchas las funcionalidades que se le podrían añadir a esta App

4.3.5. Amenazas

La amenaza de este proyecto así como todo proyecto Informatico es el continuo cambio en las tecnologías usadas, sus actualizaciones y cambios que requieren siempre un constante equipo dedicado al mantenimiento.

4.4. Análisis de Mantenibilidad, Seguridad, Disponibilidad, Usabilidad

4.4.1. Mantenibilidad

Para que la aplicación pueda ser mantenida más adelante por otro desarrollador o equipo de desarrolladores, el código ha sido debidamente comentado con aclaraciones que indican de forma clara la finalidad de cada parte de código asegurando así su fácil comprensión

4.4.2. Seguridad

Para comprobar si se cumple o no la seguridad, se ha examinado la base de datos y tomadas las medidas necesarias para evitar inyecciones SQL.

4.4.3. Disponibilidad

Por parte de la App al integrarse a corto plazo con glUCModel y al estar ésta en el PlayStore de Google, podemos asegurar que la aplicación estará disponible para su descarga en la plataforma de Google.

Por parte del servidor, al estar a cargo de nosotros, somos nosotros los que tenemos que velar porque el mismo ofrezca sus servicios el mayor tiempo posible.

4.4.4. Usabilidad

Para probar este requisito, se testeó la app continuamente en el Smartphone del autor. También se pidió la opinión de diferentes usuarios con respecto a la manejabilidad y sencillez por aprender a usar la aplicación.

Nº	Descripción	Conclusión
1	Mantenibilidad	Superado
2	Seguridad	Superado
3	Disponibilidad	Superado
4	Usabilidad	Superado

Tabla 2: Análisis de Mantenibilidad, Seguridad, Disponibilidad, Usabilidad

Capítulo 5. Manual de usuario

Menú principal

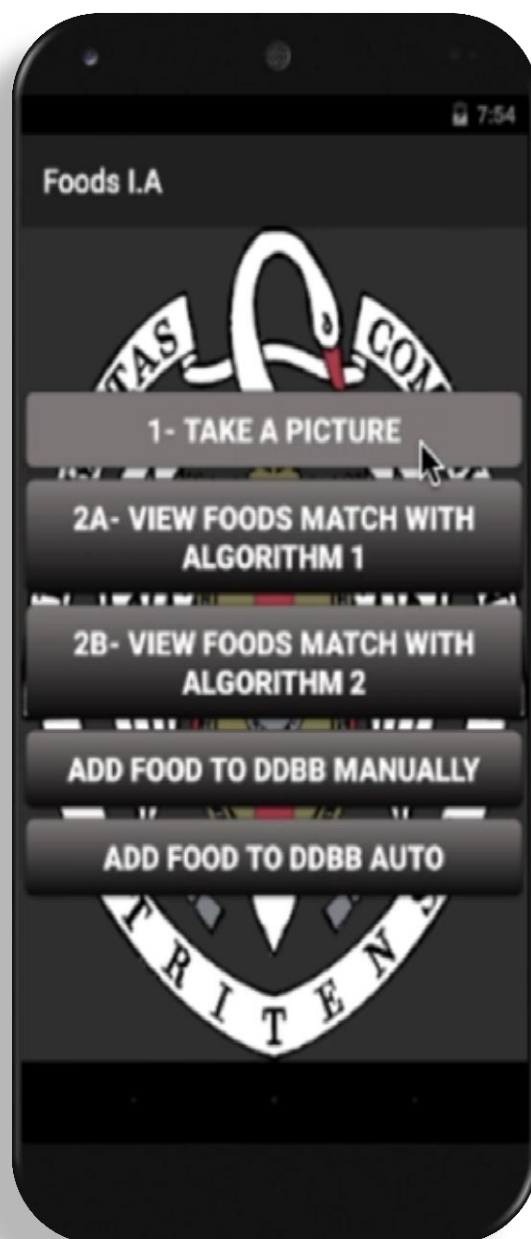


Ilustración 28: App menú principal

Usuario realiza foto (Ejemplo para un Aguacate)

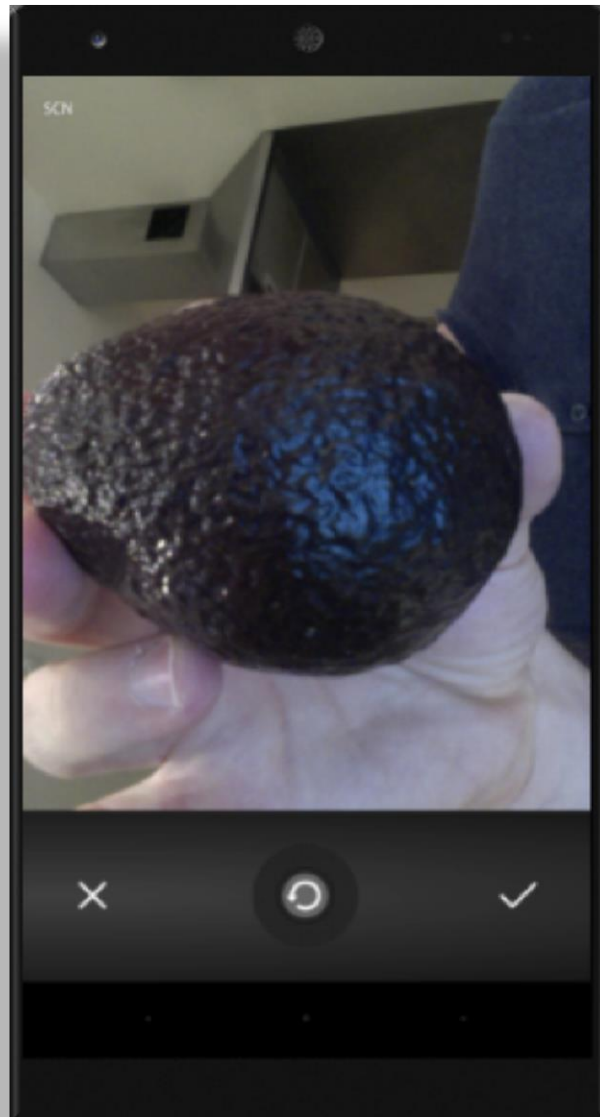


Ilustración 29: Captura de foto

Confirmación que la foto ha sido tomada y que los pixeles han sido analizados

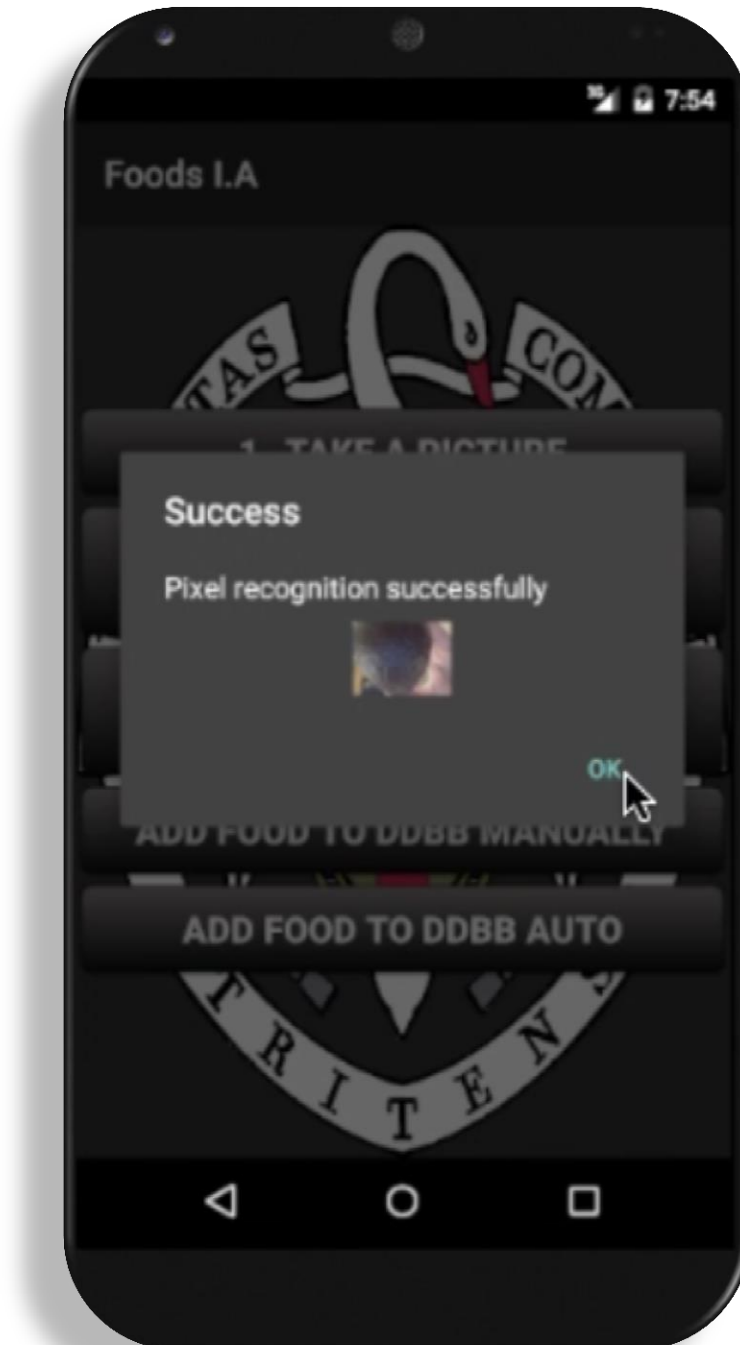


Ilustración 30: Mensajes de confirmación en la App

Se muestran posibles alimentos coincidentes



Ilustración 31: App muestra alimentos coincidentes

Se muestra mensaje de espera mientras se cargan los detalles

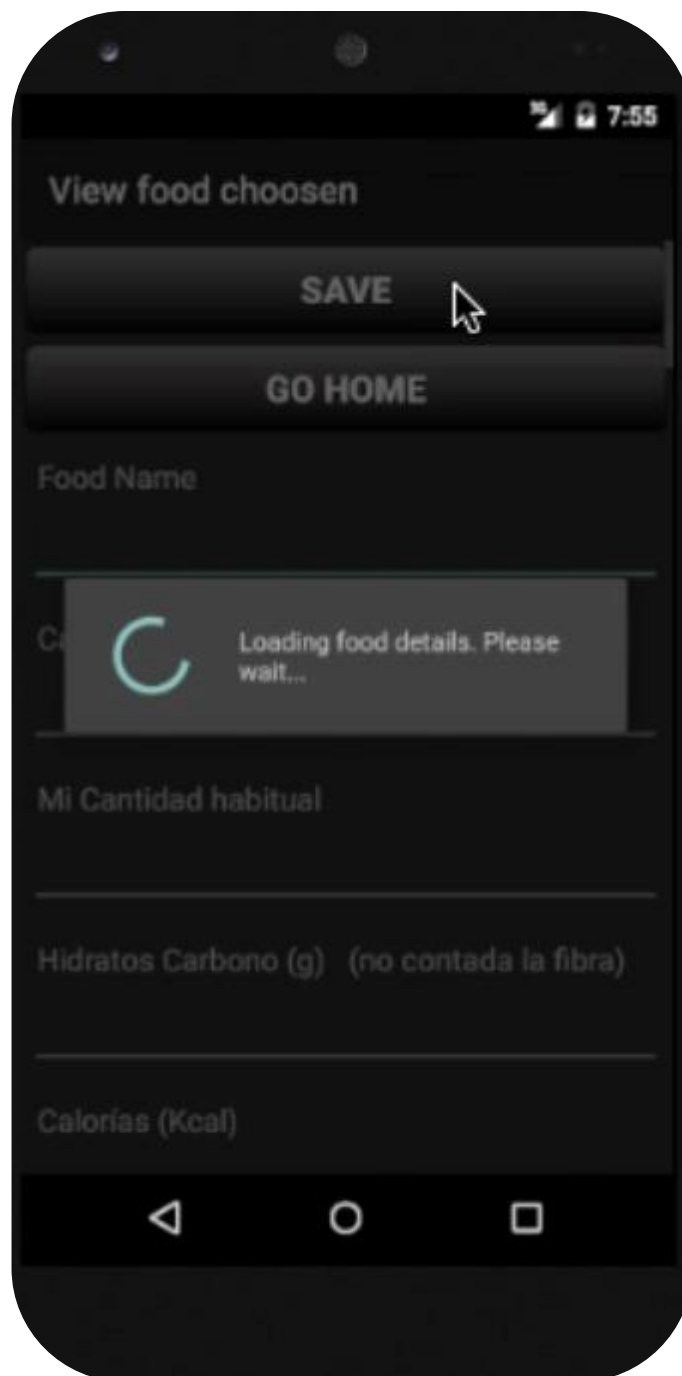


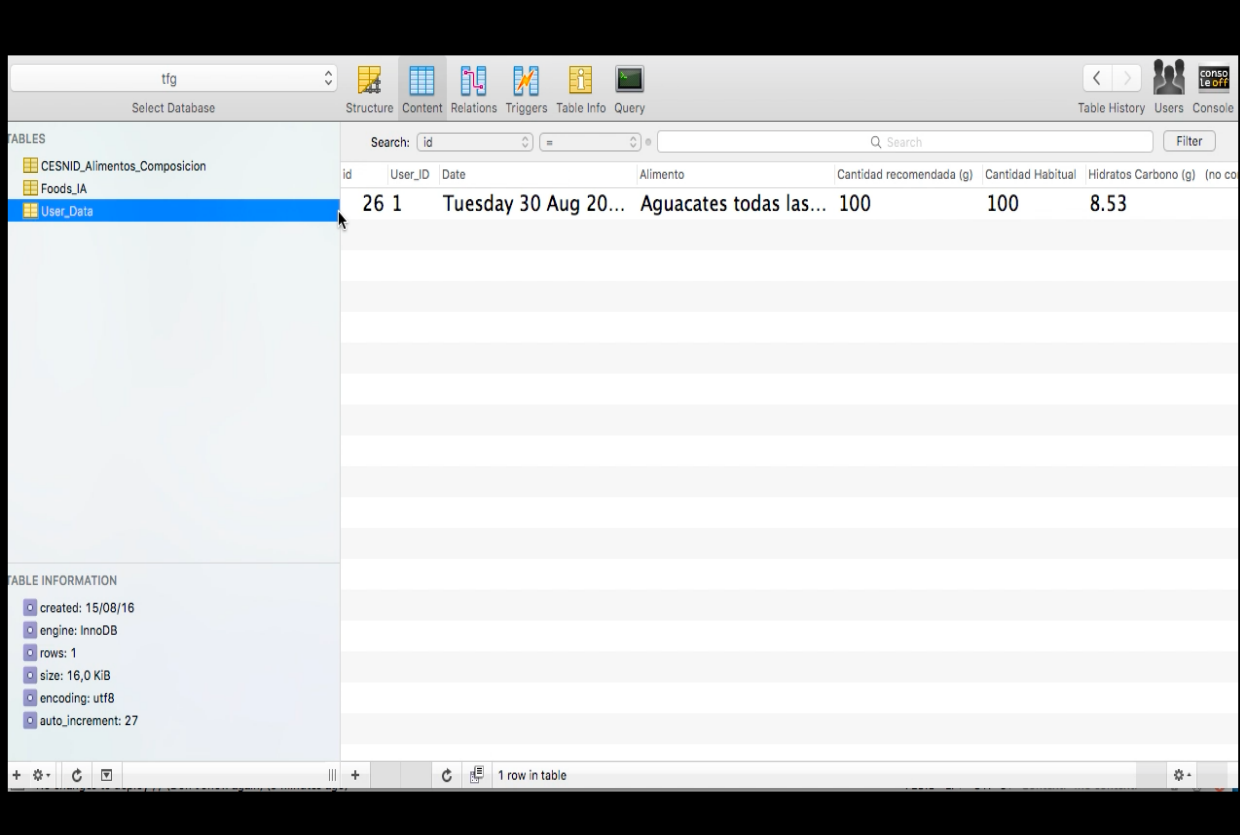
Ilustración 32: Mensajes de espera en la App

Se muestran las propiedades nutricionales del alimento seleccionado, teniendo el usuario la opción de editar el campo de la porción ingerida en gramos, actualizándose todos los demás valores en base a esta.



Ilustración 33: Propiedades nutricionales del alimento

Usuario guarda los datos del alimento y la cantidad ingerida en su registro personal.



The screenshot shows a database management interface with the following components:

- Database:** tfg
- Navigation Tabs:** Structure, Content, Relations, Triggers, Table Info, Query
- Table List:** CESNID_Alimentos_Composicion, Foods_IA, User_Data (selected)
- Table Information:**
 - created: 15/08/16
 - engine: InnoDB
 - rows: 1
 - size: 16,0 KiB
 - encoding: utf8
 - auto_increment: 27
- Table Data:**

id	User_ID	Date	Alimento	Cantidad recomendada (g)	Cantidad Habitual	Hidratos Carbono (g) (no co
26	1	Tuesday 30 Aug 20...	Aguacates todas las...	100	100	8.53
- Status Bar:** 1 row in table

Ilustración 34: Grabado de datos en la BD del usuario

En caso que no existan coincidencias, el usuario tiene la opción de agregar un nuevo alimento, para ello se hace uso de la API de USDA la cual devuelve los siguientes alimentos para el ejemplo de aguacate:

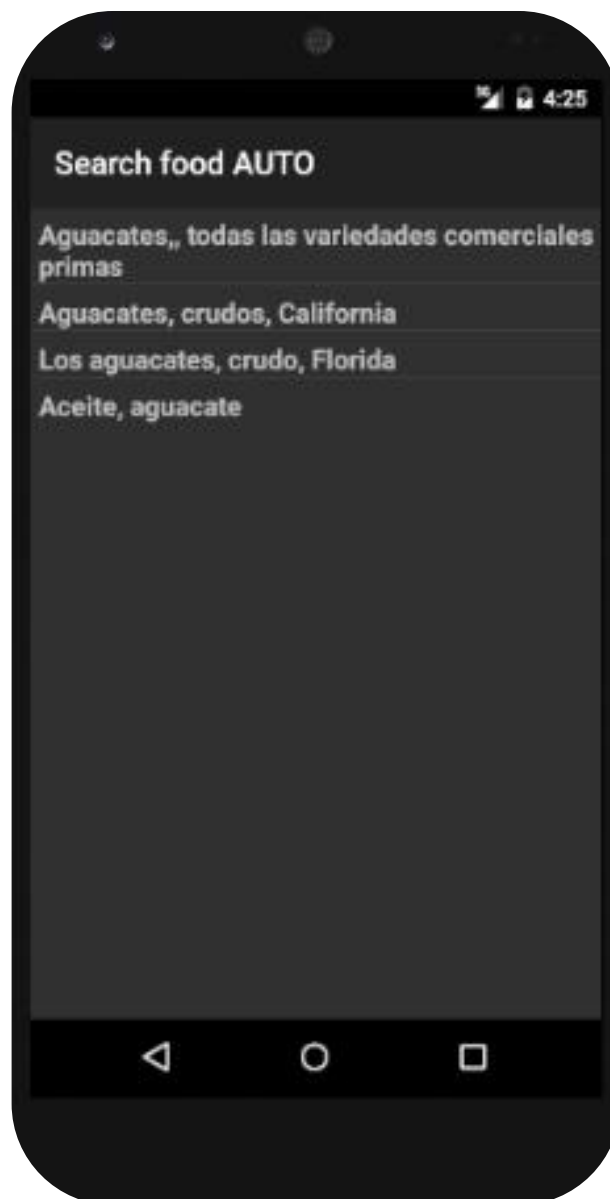


Ilustración 35: Conexión de la App con Api externa

Usuario selecciona y guarda alguno de los resultados en la BD

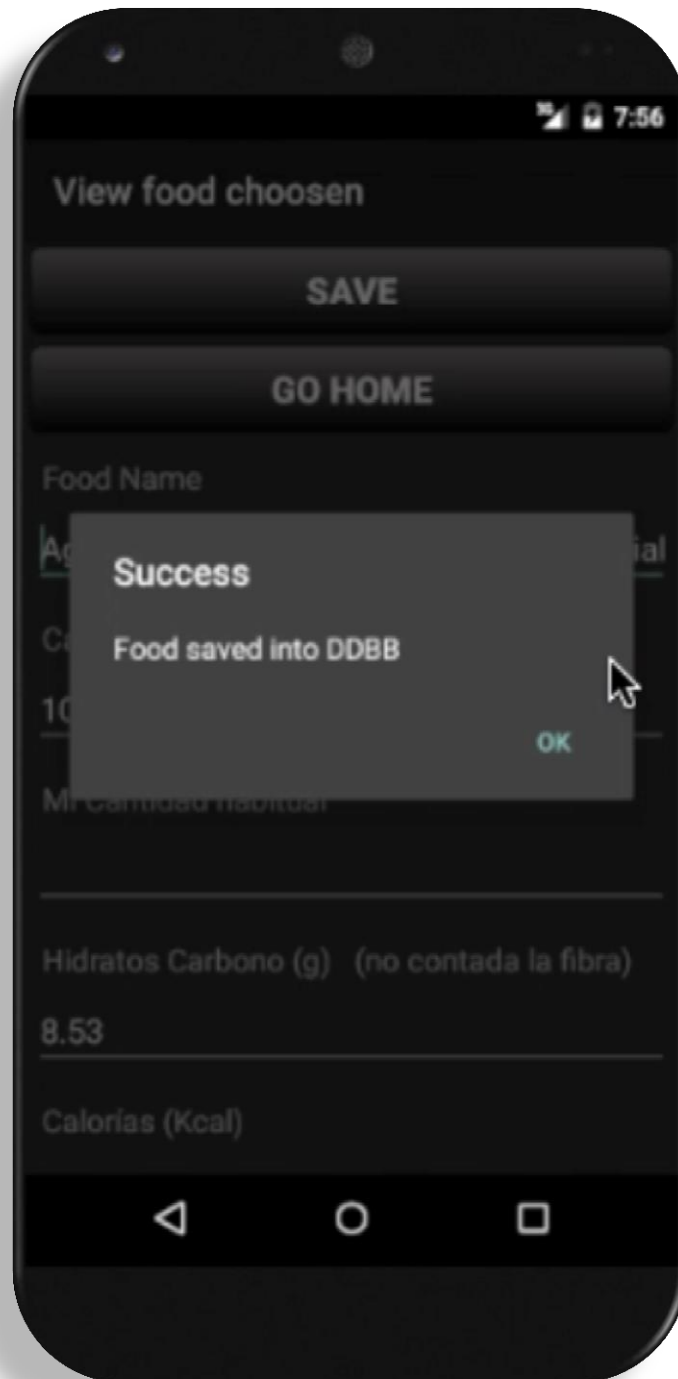


Ilustración 36: Salvado de nuevo alimento en la BD

Capítulo 6. Conclusiones y futuro de la aplicación

Se puede afirmar que tenemos una App en Android estable con una estructura cliente servidor capaz de:

- Reconocer los alimentos en la fotografía tomada .
- Mostrar más de 40 diferentes valores nutricionales para dicho alimento.
- Grabar en la BBDD del usuario la información nutricional y cantidad de alimento ingerido de forma dinámica.
- Guardar nuevos alimentos en la BBDD.

En cuanto al futuro a corto plazo se podrá implementar:

- La Integración con gIUCModel de modo que el macro proyecto siga creciendo y ofreciendo más y nuevas funcionalidades.
- Configuración del servidor backup (Ubicado en la facultad de informática con el cual se puede programar tareas de backup y respaldo de la BBDD).
- Aprovechar el servidor API para desarrollar la version en iPhone de este proyecto.
- Desarrollo del algoritmo Sobel para detección de bordes y posibles raciones, parcialmente ya implementado (Ver ilustración Sobel a continuación).

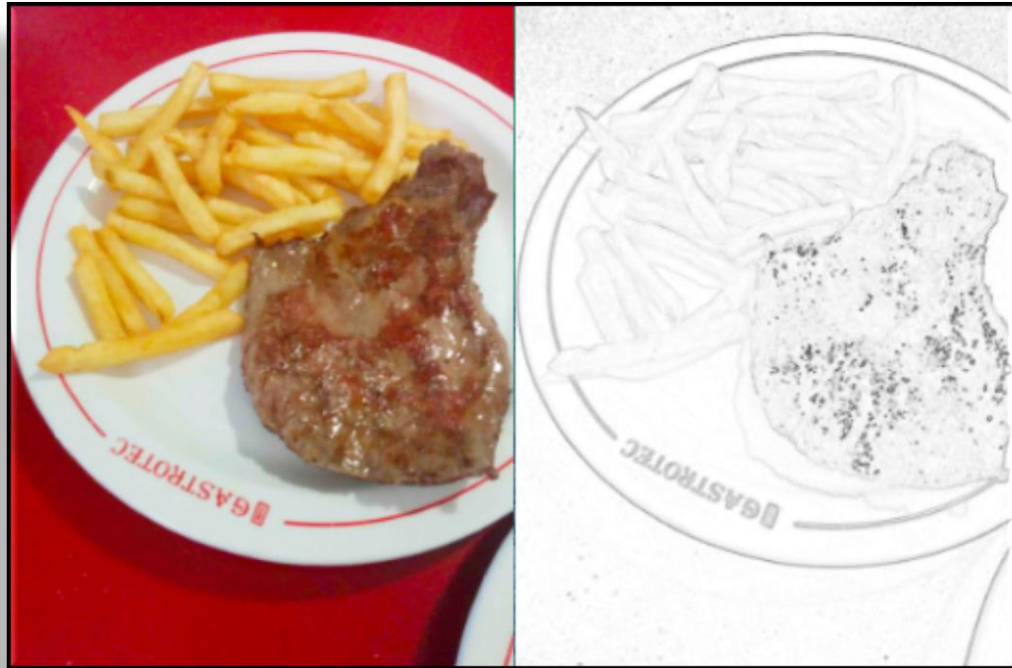


Ilustración 37: Algoritmo Sobel

Chapter 6. Conclusion and future

We can say that we have an Android App stable with a client-server structure capable of:

- Recognize food in a photograph taken.
- Show over 40 different nutritional values for that food.
- Dynamically record in the user DB the nutritional information of the food ingested.
- Save new foods in the DB.

As for the short-term future could be implemented:

- The Integration with gIUCModel so that the macro project continues to grow and offer more and new features.
- Setting The backup server (It is located in the computer science faculty with which you can schedule servers and DBs backups).
- Development Sobel algorithm for edge detection and possible portions recognition, partially already implemented (See illustration Sobel).
- Do use of the server API to develop the iPhone Version of this project.

Bibliografía

- [1] J.I. Hidalgo, et al., Modeling glycemia in humans by means of Grammatical Evolution, Appl. Soft Comput. J. (2013), <http://dx.doi.org/10.1016/j.asoc.2013.11.006>
- [2] J.I. Hidalgo, et al. glUCModel: A monitoring and modeling system for chronic diseases applied to diabetes. J Biomed Inform (2014), <http://dx.doi.org/10.1016/j.jbi.2013.12.015>
- [3] A. American-Diabetes-Association. Standards of medical care in diabetes 2010. Diabetes Care, 33(S1):11–61, 2010.
- [4] Hidalgo JI et al. glUCModel: Clarke and Parkes Error Grid Analysis of Diabetic Glucose Models obtained with Evolutionary Computation
- [5] Pajares, G. y, Santos M,(2005). Inteligencia artificial e ingeniería del conocimiento, Rama, Madrid.
- [6] Tasa de mercado de los Sistemas Operativos más usados en Smartphone. [iabspain – Market Share, Sistemas operativos], <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [7] Manual de PHP, Seguridad de Bases de Datos: Inyección de SQL, <http://www.php.net/manual/es/security.database.sqlinjection.Php>
- [8] Android version distribution, <http://www.dailytech.com/Android+Lollipop+Nears+10+Percent+Market+Penetration+After+Half+a+Year+in+the+Wild/article37333.htm>
- [10] Android developers, <https://developer.android.com/index.html>
- [11] Google translator API, <https://cloud.google.com/translate/docs/>
- [12] Food nutrition API service, <https://ndb.nal.usda.gov/>
- [13] Andrey Prikaznov: Create Your Own XML/JSON/HTML API with PHP, <http://css.dzone.com/articles/create-your-own-xmljsonhtml-English/Ingles-Page-103-of-105>
- [14] Guijarro M, Trabajo Fin de Master. Facultad de Informática. Universidad Complutense, Madrid.
- [15] Evaluation of Health-Related Quality of Life according to Carbohydrate Metabolism Status: A Spanish Population-Based Study (Di@bet.es Study),

<https://www.hindawi.com/journals/ije/2012/872305/>

<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3228950/>